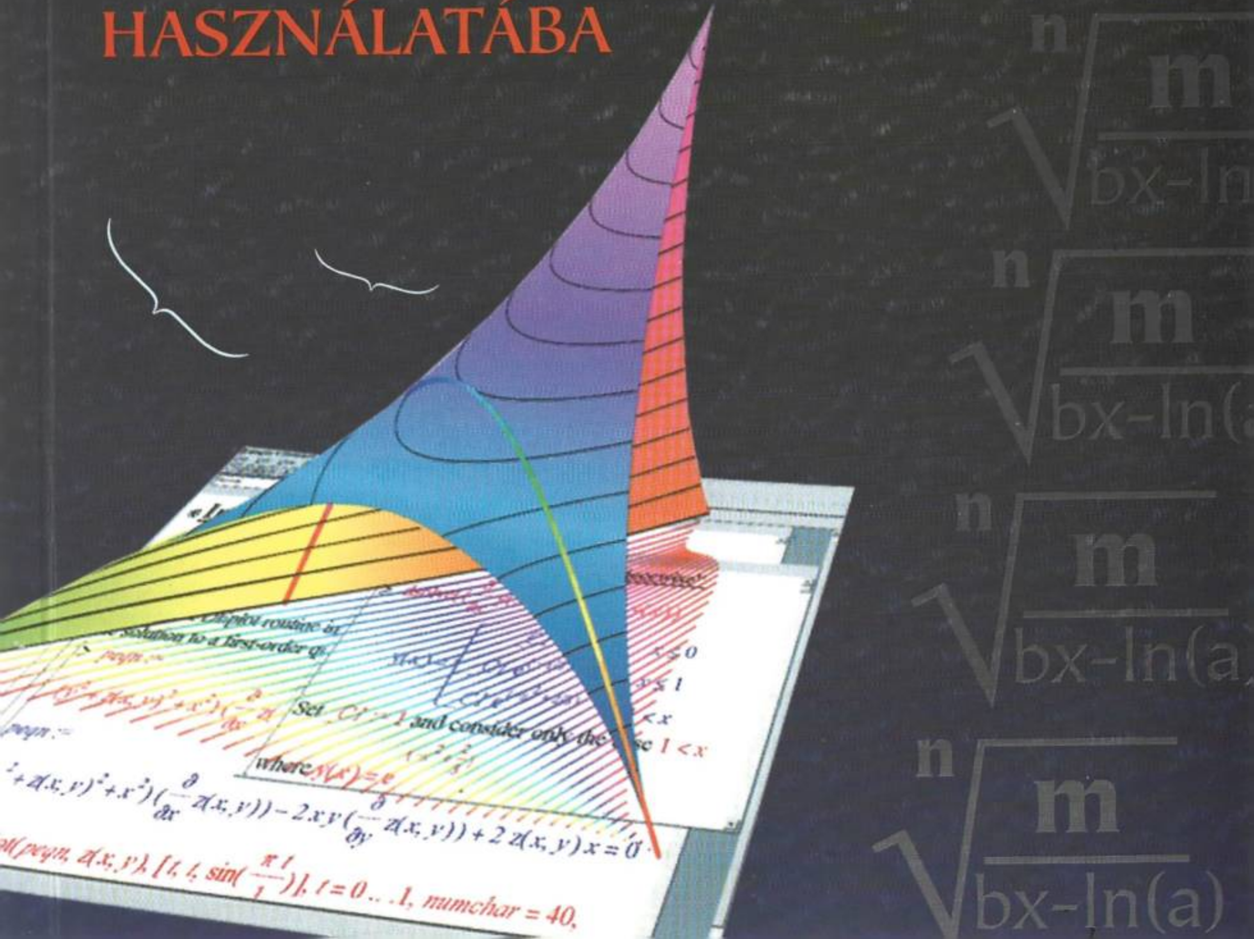
André Heck

BEVEZETÉS A Maple HASZNÁLATÁBA





Bevezetés a Maple használatába

Juhász Gyula Felsőoktatási Kiadó
Zenon Kft.
Szeged

André Heck

Bevezetés a Maple használatába

JGYF Kiadó
Szeged, 1999.

Originally published in English under the title:
Introduction to Maple by A. Heck
Copyright © Springer-Verlag New York, Inc. 1993., 1996.
All Rights Reserved.

Fordította:
Maróti György

A fordítást szakmailag ellenőrizte:
Virágh János

Nyelvi ellenőrzést végezte:
Horváthné Szélpál Mária

A Maple V Release 5 verzióhoz hozzáigazította:
Virágh János

Címlapterv:
Csécsei Károly

Nyomdai előkészítés:
Zenon Kft.

Hungarian translation copyright © Zenon Kft.

ISBN 963 9167 16 9

A második kiadás előszava

Könyvem első kiadása kedvező fogadtatásra talált a Maple-fölhasználók körében. A Maple V legfrissebb, 4-es verziója annyi új matematikai érdekességet és a fölhasználói felületen végrehajtott javítást tartalmaz, hogy a Waterloo Maple Inc. a „Power Edition" szlogennel dobta piacra. Ez a két tény tette szükségessé, hogy az első után nem sokkal egy második kiadás is elkészüljön. Kijavítottam a sajtóhibákat, néhol átfogalmaztam a szöveget, az új és átdolgozott példákon kívül több új anyagrészt is fölvettem. Alig maradt érintetlen fejezet. Új vagy lényegesen megváltozott részek: az **assume** ismertetése, az input/output, az approximáció-elmélet, az integrálás, az összetett adattípusok, az egyszerűsítés, a grafika, a differenciálegyenletek és a mátrixalgebra. A gyors referenciaként használható táblázatok a különböző sajátosságokat, a parancsok opcióit stb. összegzik. A keresés megkönnyítését szolgálja a gondosan válogatott, kibővített index. Számos új példa mutatja be, hogyan használható a Maple problémák megoldására, hogyan segíthetünk a rendszernek a számítások közben, és hogyan terjeszthetjük ki a beépített lehetőségeket.

A könyvben fölhasznált Maple verzió

A könyv második kiadását teljesen átdolgoztam a Maple V Release 4-nek megfelelően. Egész pontosan ez a kiadás a Maple V Release 4 egy SUN SPARCstation 20 71-es modellen futó 3. béta változatával készült. Nem valószínű, hogy a végleges változat ettől eltérne, bár a fölhasználói felület még kissé változhat.[1] A Maple V Release 4 ugyan sok platformon elérhető, de a könyv legnagyobb része platformfüggetlen, mivel a rendszer matematikai sajátosságaira koncentrál.

[1] A Linuxon futó Release 4-gyel készült magyar fordítás ábrái és outputjai valóban más formátumúak néhol. (*A Fordító megjegyzése.*)

A második kiadás előállításának technikai részletei

A kézírat kiszedésére a LaTeX-et használtam. A Waterloo Maple Inc. egy módosított Maple változattal segített a Maple szekciók LaTeX kóddá alakításában. A Waterloo Maple Inc. munkatársa, Stan Devitt által rendelkezésemre bocsátott eszközök lényegében ugyanúgy működnek, mint a Maple „Export to LaTeX" menüpontja. Így könnyű volt a Maple munkalapokból, az őket kísérő kommentárokból, megjegyzésekből és magyarázatokból álló szöveget összeállítani. Továbbá abban is biztos lehet az Olvasó, hogy reprodukálni tudja ezeket az eredményeket akár a terminál képernyőjén, akár papíron.

A Maple szekciók forráskódja

Az Internet-eléréssel rendelkező olvasók az összes Maple szekció forráskódját letölthetik ftp-vel az `ftp.can.nl` gépről vagy a `http://www.can.nl` WWW szerverről.

Jelölések

A Maple kulcsszavait `írógép`, a Maple eljárásokra való hivatkozásokat **félkövér** betűtípussal szedtük az egész könyvben.

Köszönetnyilvánítások

Sokaknak hálás vagyok a könyv elkészítésében nyújtott közreműködésükért. Köszönet illeti elsősorban a Maple feljesztőit. Nagyszerű eszközt hoztak létre a tudósok és a mérnökök számára. A Release 4 mind matematikai képességeit, mind könnyű használatát illetően jelentős előrelépés. Az új kiadás elkészítése ezen új lehetőségek közti izgalmas fölfedezőút volt. Nem készültem volna el ilyen hamar a Waterloo Maple Inc. munkatársai, Chris Howlett és Stan Devitt támogatása nélkül. Ellátták a program korai verzióival, a könyv írásához szükséges segédprogramokkal. Bátorítottak, mindig szívesen vették észrevételeimet, válaszoltak kérdéseimre. Külön köszönet illeti Gaston Gonnet-t, aki a zürichi ETH-n hozzáférést biztosított a Maple fejlesztői változatához, hogy lépést tudjak tartani a szoftver fejlesztésével.

Köszönöm az első kiadás olvasóitól kapott javításokat, hasznos ötleteket és elismerő szavakat. Leveleik és e-mailben küldött megjegyzéseik bátorítottak a javított és bővített kiadás megírására.

Köszönetet kívánok mondani a Springer kiadó munkatársainak, Rüdiger Gebauernek és Betty Sheehan-nek érdeklődésükért és a könyvhöz nyújtott segítségükért. Hálás vagyok a CAN Foundation támogatásáért, amely lehetővé tette, hogy egy időre megszabaduljak a CAN napi ügyeitől, és csak az új kiadáson dolgozzam.

Végül, de nem utolsósorban megköszönöm Rudi Hirschfeldnek a CWI-ből, CAN-beli kollégáimnak, Leendert van Gastel-nek és Roderik Lindbergh-nek, hogy gondosan és alaposan átolvasták a kézirat korai változatait. Értékes megjegyzéseik sokat javítottak a könyvön. Dick Verkerk a CAN Diensten-től szintén bátorított és támogatott.

Mindezen segítség ellenére könyvem olvasói bizonyára további megjegyzésekkel, javaslatokkal és korrekciókkal fognak előhozakodni. Észrevételeiket az alábbi címen várom:

University of Amsterdam
CAN Expertise Center
Attn. André Heck
Nieuwe Achtergracht 170
1018 WV Amsterdam
The Netherlands

E-mail címem

heck@can.nl

Az első kiadás előszava

A számítógépekkel végzett szimbolikus számításoknál (számítógépes algebrának is szokás nevezni) a matematikai számítások hagyományos eszközeit, a papírt és a ceruzát, a billentyűzet és a képernyő helyettesíti. A számítógépes algebrai rendszereknek nevezett interaktív programok fölhasználói a számokon kívül szimbolumokkal, formulákkal, egyenletekkel stb. is dolgozhatnak. Sok matematikai feladat, így például a differenciálás, integrálás, függvények sorfejtése, szimbolikus elemeket tartalmazó mátrixok invertálása különösebb emberi erőfeszítés nélkül gyorsan, nagy pontossággal megoldható.

A számítógépes algebrai rendszerek hathatós segédeszközei a matematikusoknak, fizikusoknak, vegyészeknek, mérnököknek, technikusoknak, pszichológusoknak, szociológusoknak — vagyis mindenkinek, aki matematikai számításokat végez. A számítógépes algebrai rendszerek nélkülözhetetlenek a modern elméleti és alkalmazott tudományos kutatásokban, valamint az oktatásban.

Ez a könyv az egyik modern számítógépes algebrai rendszert, a Maple-t ismerteti. Elsősorban arra tanít meg, hogy a Maple mire képes, és miként használható (alkalmazott) matematikai problémák megoldására. A könyv sok példát és gyakorlatot tartalmaz (elemiektől a bonyolultabbakig). Ezek a Maple használatára ösztökélnek, és a rendszerben való eligazodást segítik. Azt tanácsolom, hogy a könyvet a Maple rendszerrel együtt olvassák, próbálják ki a példák különböző variációit, és kíséreljék meg a gyakorlatok megoldását.

A hangsúlyt a Maple alapvető elveinek és ötleteinek megismertetésére helyeztük, hogy az Olvasót fölkészítsük matematikai problémáinak a rendszer segítségével történő effektív megoldására. A Maple egyes beépített lehetőségeiről az online súgó vagy a szoftverrel adott dokumentáció részletesen tájékoztat. Matematikát sem akarunk tanítani: ismertnek tételezzük föl a példák elméleti hátterét. A sokféle probléma és a Maple által nyújtott megoldások megismerése képet adhat a rendszer lehetőségeiről.

A könyvben nem tárgyaljuk kimerítően a Maple programozási nyelvét, csak egyszerű eljárásokat és alapvető nyelvi konstrukciókat használunk. Az adat-

struktúrákat azonban részletesen ismertetjük, mivel alapos megértésük nélkülözhetetlen a kifejezések hatékony kezeléséhez és egyszerűsítéséhez. Ez jó kiindulópontot jelent a pogramozási nyelvvel való további ismerkedéshez is.

A fölhasznált Maple verzió

A Maple V Release 2 verziója a nagyszámítógépektől a munkaállomásokon át a Macintosh, Next, Amiga, IBM PC és velük kompatibilis rendszerekig mindenütt elérhető. Ennek használatát tételezzük föl, bár a könyv nagyrésze platformfüggetlen.

A könyv előállításának technikai részletei

A könyv egy Silicon Graphics Indigo szerveren futó Maple V Release 2-vel készült. A Waterloo Maple Inc. által módosított program lehetővé tette az egyes Maple parancsok outputjának Postscript formátumú elmentését. Ezeket a Postscript eredményeket a TEX-hel végzett szedéskor illesztettük be a kéziratba. Így „valódi" Maple szekciókat, az őket kísérő kommentárokat, megjegyzéseket és magyarázatokat tartalmazó szöveget állítottunk össze. Ennek következtében az Olvasó is biztosan reprodukálni tudja ezeket az eredményeket akár a terminál képernyőjén, akár papíron. A Maple inputot és outputot Courier fonttal szedtük, hogy könnyen megkülönböztethető legyen a többi szövegrésztől. A Maple eljárások neveit félkövérrel írtuk. A könyv nyomdakész formátuma a CWI 1200 pont/inch fölbontású fényszedőgépén készült.

A könyv eredetéről

A szerző 1987-ben a Nijmegeni Egyetemen kezdett hozzá bevezető Maple kurzusok kidolgozásához. Azóta a kurzusok anyagának több átdolgozott és korszerűsített változata jelent meg. A legfontosabb a szintén nijmegeni Ernic Kamerichhel közösen írt 1990-es *Introductie in het gebruik van maple* című könyv. Ebben a meglévő részeket kiegészítettük, átdolgoztuk, kibővítettük. A jelen könyv alapját is ez képezi, bár a Maple V Release 2 megjelenése miatt a szöveget sok helyen meg kellett változtatni és ki kellett bővíteni. Továbbá számos, a Maple gyakorlati használatát bemutató példát is fölvettem. Mindemellett Ernic Kamerich munkája döntően hozzájárult a könyv olvashatóságához és használhatóságához.

Köszönetnyilvánítások

Sokan közreműködtek a könyv elkészítésében. Elsősorban a Waterloo-i Egyetemen a Symbolic Computation Group-ban valamint a Waterloo Maple Software-

nél dolgozó barátaimat és kollégáimat illeti köszönet támogatásukért, bátorításukért és az elmúlt néhány évben megválaszolt kérdéseimért. Rüdiger Gebauernek, a Springer kiadó munkatársának meg szeretném köszönni a könyv iránti érdeklődését és hozzám való türelmét. Hálával tartozom Darren Redfernnek és Bruce Barbernek a kézirat gondos és alapos átolvasásáért, továbbá angoltudásom és stílusom javításáért. Felbecsülhetetlen értékűek voltak Michael Monagan megjegyzései, javaslatai és kritikai észrevételei. Köszönet Ron Sommerlingnek a Maple-ről és a kurzus anyagáról folytatott beszélgetésekért. Nancy Blachman és Bert Ruitenburg a könyv egyik korai változatáról tettek megjegyzéseket. Nagyra értékelem Jan Schippernek a kézirat nyomdakész alakra hozásában nyújtott segítségét. Marc van Leeuwen a TeX használatához adott nélkülözhetetlen tanácsokat. Végül, de nem utolsósorban meg szeretném köszönni a CAN Foundation-nél és a CAN Expertise Center-ben dolgozó kollégáimnak, nevezetesen Arje Cohen-nek, Jan Sanders-nek és Leendert van Gastel-nek, hogy emlékeztettek arra, hogy a könyveket meg kell jelentetni, és intézték a CAN napi ügyeit, míg a könyvön dolgoztam. Mindezen segítség ellenére könyvem olvasói bizonyára további megjegyzésekkel, javaslatokkal és korrekciókkal fognak előhozakodni. Észrevételeiket az alábbi címen várom:

University of Amsterdam
CAN Expertise Center
Attn. André Heck
Nieuwe Achtergracht 170
1018 WV Amsterdam
The Netherlands

E-mail címem

heck@can.nl

Tartalomjegyzék

Táblázatok jegyzéke

1. Bevezetés a számítógépes algebrába

A fejezetben röviden ismertetjük a számítógépes algebrai rendszereket[1], néhány szót ejtünk történetükről, példákat mutatunk a számítógépes algebra fölhasználására, majd megvizsgáljuk ezen új technológiai eszközök előnyeit és korlátait. A fejezetet a Maple tervezésének rövid vázlatával zárjuk.

A fejezet példái néhány esetben komoly matematikai ismereteket igényelnek. A második fejezettel kezdődően részletes, lépésről lépésre haladó bemutatását adjuk a Maple rendszernek, mint szimbolikus kalkulátornak. A könyv hátralévő része különösebben nem tételezi föl a jelen fejezet ismeretét, így aki már annyira szeretné a Maple-t megismerni, hogy egy percet sem tud várni, nyugodtan átugorhatja, és később ismét visszatérhet ezekre a lapokra.

1.1. Mi a számítógépes algebra?

Történetileg a *számítani* (compute) igét leginkább a „számokkal számolni” értelemben használták. Numerikus számítások alatt azonban nemcsak a számokon

[1] A „Computer Algebra” szakkifejezés magyar megfelelőjeként a „számítógépes algebra”, ezzel összhangban a „Computer Algebra System” fordításaként a „számítógépes algebrai rendszer” terminust használjuk (*A Fordító megjegyzése.*)

végzett alapvető aritmetikai műveleteket (mondjuk az összeadást vagy a szorzást) kell érteni, hanem bonyolultabb számítási eljárásokat is, mint például matematikai függvények értékeinek numerikus kiszámítását, polinomok gyökeinek meghatározását és mátrixok sajátértékeinek numerikus közelítését. Az ilyen típusú számítások lényege, hogy aritmetikai műveleteket hajunk végre számokon és csak azokon. Továbbá a numerikus számítások az esetek nagy részében nem pontosak, mivel az alkalmazásokban majdnem mindig lebegőpontos számokat használunk. Egyszerű számításokat végre lehet hajtani papírral és ceruzával vagy zsebszámológéppel; nagy numerikus számításokra a nagy teljesítményű (mainframe) számítógépek szolgálnak. Az utóbbi ötven év a számítógépekkel végzett numerikus számítások virágkora volt; olyannyira, hogy sok tudományos kutató számára a számítógépen végzett matematikai számítás és a numerikus számítás szinonimákká váltak.

A matematikai számítások között azonban más jellegűek is előfordulnak, ezeket *szimbolikus és algebrai számításoknak* nevezhetnénk. Röviden azt mondhatjuk, hogy itt matematikai objektumokat reprezentáló szimbolumokon végzett számításoról van szó. A szimbolumok reprezentálhatnak (egész, racionális, valós, komplex vagy algebrai) számokat, de ugyanúgy jelenthetnek más matematikai objektumokat, például polinomokat, racionális függvényeket, egyenletrendszereket; vagy még absztraktabb módon matematikai struktúrákat, így csoportokat, gyűrűket, algebrákat, illetve ezek elemeit. A *szimbolikus* jelző azt hangsúlyozza, hogy a matematikai problémamegoldás végső célja sok esetben a válasz zárt alakban történő előállítása vagy valamely szimbolikus közelítésének meghatározása. Az *algebrai* alatt pedig azt értjük, hogy a számításokat a közelítő lebegőpontos aritmetika használata helyett az algebra szabályainak megfelelően pontosan hajtjuk végre. Szimbólikus algebrai számításokra példa a polinomok szorzattá alakítása, a differenciálás, az integrálás, a függvények sorbafejtése, a differenciálegyenletek analitikus megoldása, az egyenletrendszerek pontos megoldása és a matematikai kifejezések egyszerűsítése.

Az utolsó huszonöt év jelentős fejlődést hozott a szimbolikus és algebrai algoritmusok elméleti alapjainak kutatásában, továbbá a matematikai számítások számítógépeken való végrehajtására szolgáló eszközöket dolgoztak ki [19, 46, 79, 195]. Mindez egy új diszciplina létrejöttéhez vezetett, melyet számos névvel illetnek: szimbolikus és algebrai számítások, szimbolikus számítások, szimbolikus manipuláció, formula-manipuláció vagy számítógépes algebra, hogy csak néhányat említsünk. A számítógépeken végzett matematikai számításokat támogató eszközöket is legalább annyiféleképpen nevezik, mint magát a diszciplinát: szimbolikus számításokat végző programok, szimbolikus manipulátorok, számítógépes algebrai rendszerek. Sajnos a *szimbolikus számítás* szakkifejezés számos különböző kontextusban használatos a logikai programozásban, a mesterséges intelligenciában stb. Az ott használatos értelemben azonban a szimbolikus számításoknak alig-alig van köze a matematikai számításokhoz. A félreértések elkerülése végett a továbbiakban a *számítógépes algebra* kifejezést használjuk, és *számítógépes algebrai rendszerekről* fogunk beszélni.

1.2. Számítógépes algebrai rendszerek

Ebben a fejezetben a teljesség igénye nélkül nagyon röviden és szubjektíven áttekintjük a jelenleg létező számítógépes algebrai rendszereket. Alaposabb ismertetést találhatunk [42]-ben és a számítógépes algebrával foglalkozó WWW-szervereken (lásd [199]). A számítógépes algebrai rendszereket célszerű két kategóriába sorolni: a *speciális célú* és az *általános célú rendszerek* csoportjába.

A speciális célú rendszereket a fizika vagy a matematika egy adott területén fölmerülő speciális problémák megoldására tervezték. A fizikában használatosakk közül a legismertebbek a SCHOONSCHIP ([178], nagy energiájú fizika), a CAMAL ([9], égi mechanika), valamint a SHEEP és STENSOR rendszerek ([65, 104, 134], általános relativitás). A matematika területéről vett példák: a Cayley és a GAP ([28, 31, 165], csoportelmélet), a PARI, a SIMATH és a KANT ([12, 85, 103, 156], számelmélet), a CoCoA ([34, 78], kommutatív algebra), a Macaulay ([86, 173], algebrai geometria és kommutatív algebra) és a LIE ([31], Lie elmélet). Bár bennünket elsősorban az általános célú Maple rendszer [38, 39, 40, 187, 188] érdekel, a speciális célú rendszerek jelentőségét sem szabad alábecsülni, mivel rendkívül fontos szerepet játszanak a tudományos kutatás számos területén [19, 30, 106]. A speciális célú rendszerek gyakran hatékonyabbak és jobban kezelhetők az általános célúaknál. Ez az általuk használt speciális jelöléseknek és adatstruktúráknak, valamint annak a következménye, hogy algoritmusaikat többnyire alacsony szintű programozási nyelveken implementálták.

Az általános célú rendszerek változatos adatstruktúrákkal és sok matematikai függvénnyel kényeztetik fölhasználóikat, így próbálják a lehető legtöbb alkalmazási területet lefedni. (V. ö. [95].) A ma használtak közül a legrégebbi általános célú számítógépes algebrai rendszerek a MACSYMA [136] és a REDUCE [97]. Mindkét rendszer a hatvanas évek végén született, a LISP programozási nyelven implementálták őket. A MACSYMA teljes számítógépes algebrai rendszer kiegészítő csomagok széles skálájával, de számítógépes erőforrásigényei igen nagyok, és kevés platformon érhető el. A MACSYMA 1992-es PC-s verziójának megjelenésével újjászületett a rendszer. A REDUCE a nagy energiájú fizikában használatos speciális célú rendszernek készült, de fokozatosan általános célú rendszerré vált. A MACSYMA-val összehasonlítva a REDUCE szerényebb lehetőségeket kínál fölhasználóinak, másrészről azonban rendkívül nyitott rendszer (a teljes forráskód elérhető!), ami könnyen kiterjeszthetővé és módosíthatóvá teszi. A REDUCE-t továbbra is aktívan fejlesztik: a 3.6-os verziója 1995 októberében látott napvilágot. Számítógépek széles skáláján fut, és jól dokumentált.

A nyolcvanas években a PC típusú számítógépekre kifejlesztett nem programozható szimbolikus kalkulátorok első példái voltak a MuMATH [197] és követője, a DERIVE [176]. A DERIVE grafikus és numerikus lehetőségei barátságos menüvezérelt interfészen keresztül érhetők el. Kis méretét és a DOS korlátait figyelembe véve azt mondhatjuk, hogy a DERIVE a felhasználónak rendkívül izgalmas lehetőségek tárházát kínálja. Az 1994-ben kiadott 3-as verzió korlátozott programozási lehetőségeket is tartalmaz. A DERIVE sok lehetőségét beépítették a TI-92 tudományos kalkulátorba, ezzel a számítógépes algebra egészen

kicsiny gépeken is elérhetővé vált.

A legtöbb modern számítógépes algebrai rendszert a C programozási nyelven implementálták. Ezzel olyan hatékony, portábilis számítógépes programokat készíthetnek a fejlesztők, amelyek ténylegesen kihasználják a célbavett platform tulajdonságait. A számítógépes algebrai rendszerek nagy része gépek széles skáláján futtatható, a szuperszámítógépektől egészen az asztali gépekig.

Az 1.6. alfejezetben röviden összefoglaljuk a Maple [38, 39, 40, 187, 188] rendszer tervezését. Egy másik figyelmet érdemlő modern általános célú rendszer a *Mathematica* [196]. A *Mathematica* az első olyan rendszer, amelyben a szimbolikus és a numerikus számításokat, valamint a grafikai lehetőségeket úgy ötvözték egybe, hogy együttesen kellemes, felhasználóbarát környezetet kínáljanak a matematikai feladatok megoldására. A programot bizonyos gépeken (a Macintosh-on, a NeXT-en és az MS Windows-t futtató PC-ken) a jegyzetfüzetnek nevezett felhasználói felülettel látták el, amelynek segítségével közönséges szöveget, formulákat, számításokat és grafikát is tartalmazó strukturált szövegeket hozhatunk létre. A *Mathematica* másik említésre méltó tulajdonsága a jól strukturált fölhasználói szintű programozási nyelv. A megjelenését övező publicitás és marketing-stratégia (lásd [175]) kétségtelenné tette, hogy a számítógépes algebra területére is betörtek az üzleti szempontok, amelyek a rendszer képességeiről szóló kevéssé realisztikus állításokat szültek. (V. ö. [175].) Pozitív kicsengésként megemlíthetjük, hogy mindez sok kutató figyelmét felhívta a számítógépes algebrára és a számítógépes algebrai rendszerek használatára a tudományos kutatásban és az oktatásban. További előnyt jelentett a fejlesztők növekvő érdeklődése a barátságosabb felhasználói felületek tervezése, a jól használható dokumentáció és a felhasználók megfelelő támogatása iránt.

A fent említett általános célú rendszerek csak a számítógép memóriájában tárolt formulák kezelésére képesek. Így a rendelkezésünkre álló memória jelenti a formula nagyságára az egyetlen korlátot. A FORM [99, 148, 184] nevű szimbolikus manipulációkat végző programot úgy tervezték, hogy „virtuálisan végtelen” nagyságú formulákat is kezelni tudjon. A rendszerben rendelkezésre álló utasítások halmaza viszont eléggé korlátozott.

A Cayley leszármazottja az 1994-ben elkészült Magma [17, 18, 32], melynek tervezését olyan algebrai fogalmakra alapozták, mint a struktúrák és a morfizmusok. Algebrai, számelméleti, geometriai és algebrai kombinatorikai számítások támogatására szánták. Ennek eléréséhez bőséges eszköztárral látták el a rendszert a csoportok, gyűrűk, modulusok, algebrák, geometriai struktúrák és véges véletlen struktúrák (kódok, gráfok) kezelésére. Két tervezési alapelvük a modern algebrából kölcsönzött szigorú típusolási séma (melyben a típusok algebrai struktúráknak felelnek meg) és az algoritmikus valamint az adatbázisokra vonatkozó ismeretek integrációja volt.

A MuPAD [67,68] neve a „Multi Processing Algebra Data Tool” kifejezésből származik. Szimbólikus és numerikus számításokat, párhuzamos matematikai programozást és matematikai vizualizációt támogató rendszer. Nem üzleti jellegű fölhasználásra szabadon terjesztik. A Paderborni Egyetemen most is aktívan fejlesztik, a rendszer matematikai tudása gyorsan növekszik. Az 1.2.2 verzió

1995 júliusában jelent meg.

Létezik párhuzamos szimbolikus számításokat végző, a Maple-re épülő portábilis rendszer is. Neve ||Maple|| (vagyis „párhuzamos Maple"), lásd [168]. A rendszer a Strand [64] párhuzamos deklaratív nyelv és a Maple szekvenciális számítógépes algebrai rendszer közti interfészként működik, s ezzel a Strand eleganciája mellett a Maple-ben megírt hatékony szekvenciális algoritmusokat is elérhetővé teszi.

Végül, de nem utolsósorban megemlítendő az AXIOM [54, 55, 108, 109, 179]. Ezt a hatékony általános célú rendszert a nyolcvanas években az IBM Thomas J. Watson Kutatólatoratóriumában „Scratchpad" néven fejlesztették ki. A legtöbb általános célú rendszerrel ellentétben, amelyek csak előre meghatározott algebrai tartományokban, például a racionális számtestben vagy az egészek fölütti polinomok gyűrűjében végzett számításokat engednek meg, az AXIOM fölhasználói különféle típusú új algebrai struktúrákat definiálhatnak és ezekben is számolhatnak. Az 1995-ös 2.0 verzió csak nagy IBM RS/6000, SUN és HP9000/700 gépeken érhető el.

1.3. A számítógépes algebrai rendszerek tulajdonságai

Az egyes számítógépes algebrai rendszerek természetesen különbözőek, de ugyanakkor számos közös tulajdonságuk is van. A közös tulajdonságokat a Maple-ből vett példákkal illusztráljuk.

A számítógépes algebrai rendszerek olyan *interaktív* programok, amelyek a numerikus számítógépes programokkal szemben *szimbolikus* kifejezésekkel való matematikai számításokat is megengednek. Tipikus használatuk: a felhasználó beír egy matematikai kifejezést vagy utasítást, melyet azután a számítógépes algebrai rendszer megpróbál végrehajtani. A számítás eredményét megkapva a fölhasználó újabb parancsot adhat ki. Ez a folyamat egy termékeny számítógépes algebrai szekció[2] létrejöttéhez vezethet. A Maple fölhasználásának egyszerű példájaként kiszámoljuk az

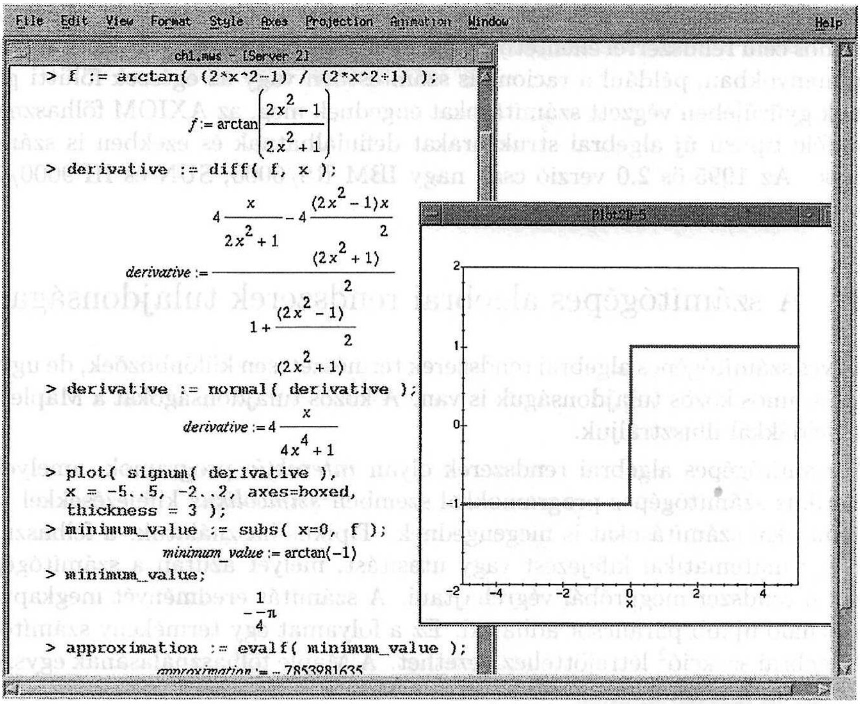
$$x \mapsto \arctan\frac{2x^2-1}{2x^2+1}$$

racionális függvény stacionárius helyét és a függvény értékét ebben a pontban. A fejezet későbbi részében látni fogjuk, hogy a Maple maga is ki tudja számítani a függvény minimumát. A példát csupán a számítógépes algebrai rendszerek néhány jellemző tulajdonságának bemutatására használjuk. Az 1.1. ábrán látható a számításokat tartalmazó munkalap, amelyet a Maple V Release 4 Unixos gépen futó munkalapos felhasználói felületű változatával futtatunk.

Vegyük közelebbről szemügyre ezt a példát. Amikor a Maple-t az X Window rendszer alatt az **xmaple** paranccsal indítjuk el, a képernyőn egy üres munkalap jelenik meg. Első sorában a Maple a „>" karakterrel jelzi, hogy parancsra vár. Ezt a „>" jelet a továbbiakban *Maple promptnak* nevezzük.

[2] A „session" kifejezést jobb híján szekciónak fogjuk mondani. (*A Fordító megjegyzése.*)

Az első paranccsal f képletét adtuk meg. Az input sort pontosvesszővel fejeztük be, utána megnyomtuk a RETURN[3] gombot. A két utolsó karakter jelzi a Maple-nek, hogy kezdheti a munkát. A beírt képlet kétdimenziós matematikai jelöléssel, a tankönyvekből ismert formában és minőségben jelenik meg. Leginkább az lepheti meg az Olvasót, hogy a rendszer megengedi az x-hez hasonló szimbolumok használatát. A legtöbb numerikus programozási nyelvben ez azonnali hibaüzenetet okozna; nem így a *szimbolikus* számításokat végző rendszerekben!



1.1. ábra: Egy Maple munkalap

A Maple minden utasítás végrehajtása után új prompt jelet ír ki, jelezvén, hogy a következő parancsra vár. Az f-et függvényként kívánjuk fölfogni, **diff**erenciáltatjuk a **diff** paranccsal, és az eredményt a *derivative* nevű változóhoz rendeljük hozzá. A Maple válasza meglehetősen bonyolult kifejezés, így **normal**izáljuk a racionális függvényt a **normal** parancs segítségével. A válasz olyan egyszerű kifejezés, amelynek előjele könnyen kirajzoltatható. Az ábra alapján azonnal látható, hogy az eredeti függvénynek minimuma van az $x = 0$ helyen. A minimum értékét, $-\frac{\pi}{4}$-et, az $x = 0$ érték f-be való behelyettesítésével, vagyis a **subs** parancs alkalmazásával kapjuk. Az **evalf** parancs ezen érték lebegőpontos közelítését adja. Az **evalf** elnevezése (az „**eval**uate using **f**loating-point arith-

[3] A PC billentyűzetén ehelyett általában az ENTER-t találjuk. (*A Fordító megjegyzése*)

metic" kifejezés rövidítése) immár a negyedik példáját adja a Maple névadási filozófiájának: válasszunk rövid, könnyen megjegyezhető, az eljárások funkciójára utaló neveket. Saját változóinkat mi is „értelmes", használatukra utaló nevekkel láttuk el.

A Maple láthatóan nem törődik azzal, hogyan ismerjük ki magunkat az elvégzendő számítás részletei között. A második eredmény alapján *nekünk* kell eldönteni, hogy megpróbálunk kevésbé bonyolult formulát találni a függvény deriváltjára. Az Olvasó megkérdezhetné, hogy a rendszer vajon miért nem hajtja végre ezt a többé-kevésbé nyilvánvaló egyszerűsítést. De ne feledjük, nem mindig nyilvánvaló, mit és hogyan egyszerűsítsünk. Sokszor több lehetséges egyszerűsítés közül a matematikai környezet dönti el, hogy melyik a megfelelő. Például az

$$\frac{(x^2-1)(x^2-x+1)(x^2+x+1)}{(x-1)^6}$$

racionális kifejezés a kompaktabb

$$\frac{x^6-1}{(x-1)^6}$$

alakra hozható, de integráláshoz célszerűbb így fölírni:

$$1+\frac{6}{(x-1)^5}+\frac{15}{(x-1)^4}+\frac{20}{(x-1)^3}+\frac{15}{(x-1)^2}+\frac{6}{x-1}.$$

Az automatikus egyszerűsítéssel kapcsolatos másik probléma az, hogy sok számításnál nem lehet előre megjósolni az eredmény nagyságát és alakját, ezért bármikor képesnek kell lennünk a beavatkozásra. Egy olyan eljárás, amely az egyik esetben kiválóan működik, máskor még nagyon rossz választásnak bizonyulhat. Például azt gondolhatnánk, hogy a kifejezések szorzattá alakítása mindig jó ötlet. Az

$$x^8+8x^7+28x^6+56x^5+70x^4+56x^3+28x^2+8x+1$$

formula szorzattá alakításának eredménye

$$(x+1)^8.$$

Ugyanakkor azon kívül, hogy mennyire költséges, még igencsak meglepő is az aránylag egyszerű

$$x^{26}+x^{13}+1$$

formula szorzattá alakításának eredménye:

$$(x^{24}-x^{23}+x^{21}-x^{20}+x^{18}-x^{17}+x^{15}-x^{14}+x^{12}-x^{10}+x^9-x^7+$$
$$x^6-x^4+x^3-x+1)(x^2+x+1)$$

Ezen megfontolásokból a Maple csak akkor alkalmaz automatikus egyszerűsítési szabályokat, ha a keletkező kifejezés kétségtelenül egyszerűbb. Az $x+0$ kifejezést nyilván x-re kell egyszerűsíteni, a $3x$ egyszerűbb, mint $x+x+x$, az x^3

jobb kifejezés, mint $x \times x \times x$ és $sin(\pi)$ is 0-ra egyszerűsítendő. Bármely más egyszerűsítés a felhasználó feladata. A Maple mindössze az eszközöket biztosítja az ilyen feladatok elvégzéséhez.

Az automatikus egyszerűsítés bizonyos esetekben a matematikai korrektség elvesztésével jár. Például a $0 \times f(1)$ kifejezés automatikus egyszerűsítése 0-ra nem mindig helyes. Ilyen kivételes esetek, amikor $f(1)$ definiálatlan vagy végtelen. Az automatikus egyszerűsítés végrehajtását csak akkor kívánnánk, ha $f(1)$ véges, de nehéz kiszámítani. Hasonló esetekben a számítógépes algebrai rendszerek tervezőinek választani kell a szigorú matematikai helyesség, valamint a rendszer használhatósága és hatékonysága között. (V. ö. [60].) A Maple és sok más rendszer esetében a mérleg nyelve néha kicsit elbillen a hatékonysági okokból alkalmazott, nem 100%-ig biztonságos egyszerűsítési eljárások irányába.

Az első példánkból látható másik említésre méltó tény az, hogy a Maple az origóban a függvény pontos értékét, $-\frac{1}{4}\pi$-t veszi, nem pedig valamilyen közelítő értéket, mondjuk 0.785398-ot. Ez a számítógépes algebrai rendszerek második fontos aspektusa: a *pontos aritmetika* hangsúlyozása. A számítógépes algebrai rendszerek képesek a *fölhasználó által meghatározott pontosságú* lebegőpontos aritmetikában számolni. Így például a Maple-ben a $\tan^2(\pi/12)$ kifejezés értékét pontosan meghatározhatjuk, de megkaphatjuk 25 tizedesjegy pontosságú valós lebegőpontos alakú közelítését is:

```
> real_number := tan(Pi/12)^2;
```

$$real_number := (2-\sqrt{3})^2$$

```
> real_number := expand( real_number );
```

$$real_number := 7-4\sqrt{3}$$

```
> approximation := evalf( real_number, 25 );
```

$$approximation := .07179676972449082589021 6$$

A Maple-höz hasonló számítógépes algebrai rendszerek jelentős mennyiségű beépített matematikai tudásanyaggal bírnak. Ez teszi őket hasznos matematikai segítőtársakká. Az analízisben elvégzik a függvények differenciálását, határértékeket és sorfejtéseket határoznak meg. A számítógépes analízis egyik fontos területe a (határozatlan és a határozott) integrálás. A Maple elemi függvények integrálására nem a klasszikus algoritmusokat, hanem az ún. Risch algoritmust [76] használja, szemben a legtöbb matematikai kézikönyvben ismertetett heurisztikus integrálási eljárásokkal.

A rendelkezésre álló függvénytani eszközökkel könnyen felderíthetjük a függvények tulajdonságait. A következő Maple szekcióban az előzőleg definiált f függvényt vizsgáljuk. A „#" karakterrel kezdődő sorokat a rendszer kommentároknak tekinti; a jól megválasztott változónevekkel és a Maple által használt „kifejező" eljárásnevekkel együtt így általában könnyen elmagyarázható a munkalap tartalma. Ha az input sort pontosvessző helyett kettősponttal zárjuk, akkor a Maple nem nyomtatja ki a kapott eredményt. Szóközök használata a parancssorban opcionális, de a jobb olvashatóság kedvéért néhol érdemes használni őket.

```
> f := arctan( (2*x^2-1) / (2*x^2+1) ); # enter formula
```

$$f := \arctan(\frac{2x^2-1}{2x^2+1})$$

```
> df := normal( diff( f, x ) ); # differentiate f
```

$$df := 4\frac{x}{4\,x^4+1}$$

```
> F := integrate( df, x ); # integrate derivative
```

$$F := \arctan(2\,x^2)$$

```
> normal( diff( F - f, x ) ); # verify F = f + Pi/4
```

$$0$$

```
> eval( subs( x=0, F - f ) );
```

$$\frac{1}{4}\pi$$

```
> readlib( extrema ):  # load library function
> extrema( f, {}, x, stationary_points );
```

$$\{-\frac{1}{4}\pi\}$$

```
> stationary_points;
```

$$\{\{x = 0\}\}$$

```
> # this value was assigned by the call of 'extrema'
> solve( f=0, x ); # compute the zero's of f
```

$$\frac{1}{2}\sqrt{2},\ -\frac{1}{2}\sqrt{2}$$

```
> # compute the Taylor series approximation of f
> series( f, x=0, 15 );
```

$$-\frac{1}{4}\pi+2\,x^2-\frac{8}{3}x^6+\frac{32}{5}x^{10}-\frac{128}{7}x^{14}+\mathrm{O}(x^{15})$$

```
> # load package for numerical approximation of functions
> with( numapprox ):
> pade( f,x, [6,4] );
```

$$\frac{1}{12}\frac{-15\,\pi+120\,x^2-36\,\pi\,x^4+128\,x^6}{5+12\,x^4}$$

```
> # compute the Chebyshev-Pade approximation of f
> chebpade( f, x, [2,2] );
```

$$\frac{-.007904471007\,\mathrm{T}(0,x)+.4715125862\,\mathrm{T}(2,x)}{\mathrm{T}(0,x)+.4089934686\,\mathrm{T}(2,x)}$$

```
> # compute limit of f when x goes to infinity
> limit( f, x=infinity );
```

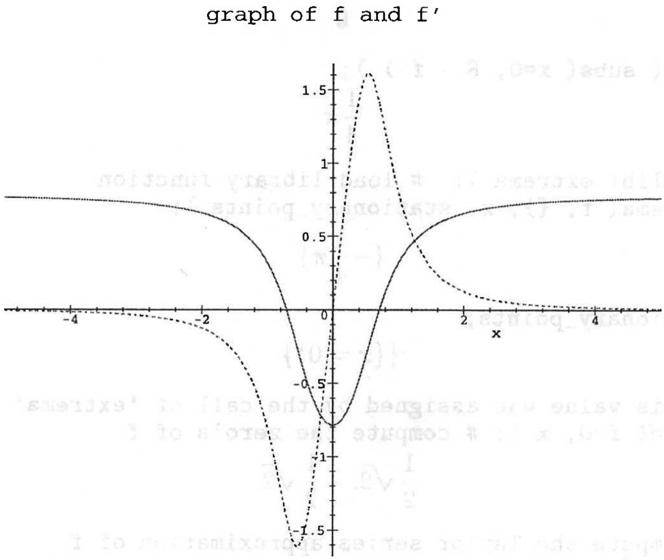
$$\frac{1}{4}\pi$$

```
> # compute the asymptotic form of f
> series( f, x=infinity );
```

$$\frac{1}{4}\pi-\frac{1}{2}\frac{1}{x^2}+\mathrm{O}(\frac{1}{x^6})$$

```
> # finally, draw the graphs of f and df
> f_plot := plot( f, x=-5..5, linestyle=0 ):
> df_plot := plot( df, x=-5..5, linestyle=4 ):
> plots[display]( {f_plot, df_plot},
> title='graph of f and f'' );
```

A függvények grafikonja az 1.2. ábrán látható.



1.2. ábra: Az $(x, y) \mapsto \arctan\left(\frac{2x^2-1}{2x^2+1}\right)$ függvénynek és deriváltjának grafikonja

A számítógépes algebra más látványos területei a polinomkalkulus, a lineáris és nem lineáris egyenletrendszerek megoldása, a rekurrens és a differenciálegyenletek megoldása, műveletek numerikus és szimbolikus komponensekből álló mátrixokkal, végül a tenzorkalkulus. Számos formula-manipulációs eszköz áll rendelkezésünkre például részkifejezések kiválasztására és helyettesítésére, korlátozott egyszerűsítésre, egyszerűsítési szabályok megadására, mintaillesztésre és így tovább. A számítógépes algebrai rendszereket olyan *matematikai szakértői rendszereknek* is nevezhetnénk, amelyekkel a matematikai problémákat hatékonyabban és pontosabban oldhatjuk meg, mint papírral és ceruzával.

Azon túl, hogy szimbolikus és algebrai kalkulátorként használjuk őket, a legtöbb számítógépes algebrai rendszert programozási nyelvként is alkalmazhatjuk új matematikai algoritmusok implementálására. Ezt illusztrálandó írunk egy olyan Maple programot, amely az $y_n(x)$ Bessel polinomok értékeit számítja ki. Idézzük föl [90]-ből a megfelelő rekurzív definíciókat:

$$\begin{aligned} y_0(x) &= 1, \\ y_1(x) &= x + 1, \\ y_n(x) &= (2n-1)\, x\, y_{n-1}(x) + y_{n-2}(x), \quad \textbf{ha } n > 1. \end{aligned}$$

```
> Y := proc( n::nonnegint, x::name )
>    if n=0 then 1
>    elif n = 1 then x+1
>    else Y(n,x) := expand( (2*n-1)*x*Y(n-1,x) + Y(n-2,x) )
>    fi
> end:
> Y(5,z);
```

$$945\,z^5 + 945\,z^4 + 420\,z^3 + 105\,z^2 + 15\,z + 1$$

A Maple programozási nyelve a deklarációk nélküli Algol68-ra emlékeztet, de több funkcionális programozási paradigmát is tartalmaz.

1.4. A számítógépes algebra előnyei

A számítógépes algebra hosszú távú célja a matematikai problémák megoldási folyamatának minél nagyobb arányú automatizálása. Bár a mai számítógépes algebrai rendszerek még messze nem képesek a problémák automatikus megoldására, így is hasznos, sőt talán elengedhetetlen kutatási és oktatási segédeszközök. Természetesen a számítógépes algebrai rendszerekkel való megismerkedéshez idő kell, de ez az idő jól kamatozik. Ebben a részben a Maple-t használó példákkal illusztráljuk az ilyen rendszerek megismerésének és alkalmazásának néhány fontosabb indítékát. Némelyik példa mélyebb matematikai ismereteket igényel. Az összes számolást 75 Mhz-es Super SPARC II processzoros, 64 Mbyte memóriával és 128 Mbyte swap területtel ellátott, SunOS 4.1.3 operációs rendszerű SUN SPARCstation 20 M71 típusú munkaállomáson végeztük.[4] Ez nem azt jelenti, hogy sokkal kisebb gépen nem lehetett volna ugyanezeket az eredményeket megkapni; legföljebb több lett volna az elhasznált gépidő.

A számítógépes algebrai rendszerek fő előnye, hogy terjedelmes szimbolikus számítások végzésére képesek. Bár sokszor papírral és ceruzával kiszámolható triviális standard átalakításokról van szó, minél nagyobbak a formulák, annál nehezebb a feladat, és annál kisebb a sikeres végrehajtás esélye. Az ilyen jellegű problémáknál nagyszerű segédeszköz egy számítógépes algebrai rendszer.

Első példánk az R. Pavelle által [154]-ben fölvetett problémák egyike, melyeket a számítógépes algebrai rendszerek számára kihívásként fogalmazott meg. A feladat a következő: bizonyítsuk be, hogy

$$\frac{\sin\left(\dfrac{nz\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+z^2}}\right)}{\sqrt{x^2+y^2+z^2}}$$

[4] A fordításban szereplő eredményeket a Maple V Release 4 Linuxos gépen futó változatával nyertük. Ezen készült a fordítással kapcsolatos összes szerkesztési, grafikai munka is. A gép paraméterei: 166 Mhz-es Intel Pentium processzor, 32 Mbyte memória, 128 Mbyte swap, Redhat 5.0 disztribució 2.0.33 kernellel. (*A Fordító megjegyzése.*)

megoldása a következő negyedrendű parciális differenciálegyenletnek

$$\left(\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)+n^2\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\right)f=0.$$

A Maple egyszerűsítési eljárásai elég hatékonyak ahhoz, hogy ezt a problémát másodpercek alatt megoldja:

```
> settime := time(): # start timing
> f := sin( n*z*sqrt(x^2+y^2+z^2) / sqrt(y^2+z^2) ) /
> sqrt(x^2+y^2+z^2);
```

$$f:=\frac{\sin\left(\frac{n\,z\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+z^2}}\right)}{\sqrt{x^2+y^2+z^2}}$$

```
> simplify( diff( diff(f,x$2) + diff(f,y$2) + diff(f,z$2),
>   x$2 ) + n^2 * ( diff(f,x$2) + diff(f,y$2) ) );
```

$$0$$

```
> cpu_time = (time()-settime) * seconds; # computing time
```

$$cpu_time = 11.460\,seconds$$

Második példánk célja a G_2 típusú Lie csoport reprezentációjának dimenzióit előállító generátor-függvény meghatározása. (V. ö. [43, 44].) Tehát olyan $F(x,y)$ racionális függvényt próbálunk találni, amelyre

$$F(x,y)=\sum_{k,l\geq 0} G2(k,l)x^k y^l,$$

ahol G2(k,l) a következő kifejezés:

```
> G2 := (k,l) -> 1/5! * (k+1) * (l+1) * (k+l+2) *
>   (k+2*l+3) * (k+3*l+4) * (2*k+3*l+5);
```

$$G2 := (k,l) \to \frac{1}{120}(k+1)(l+1)(k+l+2)(k+2\,l+3)(k+3\,l+4)(2\,k+3\,l+5)$$

Itt a Maple nyíl jelölését használtuk a függvényoperátorokra. Ily módon $G2$ argumentumainak segítségével megadott értékeket fölvevő függvény, nem pedig egyszerű formula. Racionális generátorfüggvények a Maple **genfunc** csomagjával kezelhetők. Ezt használjuk problémánk megoldására is:

```
> with( genfunc ): # load genfunc package
> settime := time(): # start timing
> F := rgf_encode( rgf_encode( G2(k,l), k, x), l, y ):
> F := sort( factor( F ) );
```

$$F := (x^4y^4+8x^4y^3+x^3y^4+8x^4y^2-26x^3y^3+x^4y-41x^3y^2 +15x^2y^3-6x^3y+78x^2y^2-6xy^3+15x^2y-41xy^2+y^3 -26xy+8y^2+x+8y+1)/((x-1)^6(y-1)^6)$$

```
> cpu_time = (time()-settime) * seconds; # computing time
```

$$cpu_time = .709\, seconds$$

A tudományos életből is említünk egy olyan példát, melyben a Maple a matematikai segédeszköz szerepét játszhatta volna (lásd [198]). Ebben a cikkben a Δ Laplace–Beltrami operátor meghatározására volt szükség hiperszférikus koordinátarendszerben. Ehhez ki kellett számolni a koordinátafüggvények Jacobi-mátrixát, a metrikus tenzort és ennek inverzét. Idézünk néhány mondatot a cikkből:

> „Nem nehéz kiszámítani $\dfrac{\partial \mathbf{Y}}{\partial q_i}$-t és szintén nem túl nehéz, de elég unalmas számolással adódnak a (32B) egyenlet nyomai. Némi algebra bevetése után azt kapjuk, ...”
>
> „A g függvény invertálása is elég fáradságos munka. Néhány oldalon kiszámolva a minorokat, kijön, hogy ...”

Ezek a megjegyzések valóban igazak, ha a számításokat papírral és ceruzával végezzük; más a helyzet, ha a Maple-re bízzuk a feladat végrehajtását. Az alábbi Maple számításokban a lehetőségek határai között megtartottuk a [198] cikk eredeti jelöléseit. Ne zavarjon, ha nem értjük minden egyes parancs jelentését, a részleteket megtalálhatjuk a könyv későbbi fejezeteiben.

A számítás első lépése az Y koordináta-leképezés definiálása és a G metrikus tenzor megkonstruálása. Kiderül, hogy ez a számítás legidőigényesebb része:

```
> settime := time(): # start timing
> with( linalg ): # load linear algebra package

Warning, new definition for norm
Warning, new definition for trace

> R[z] := x -> array( [ [ cos(x), -sin(x), 0 ],
>   [ sin(x), cos(x), 0 ],
>   [ 0, 0, 1 ] ] ):
> 'R[z](phi)' = R[z](phi);
```

$$R_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> R[y] := x -> array( [ [ cos(x), 0, -sin(x) ],
>   [ 0, 1, 0 ],
>   [ sin(x), 0, cos(x) ] ] ):
> 'R[y](phi)' = R[y](phi);
```

$$R_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

```
> T := x -> array( [ [ cos(x) + sin(x), 0, 0 ],
>     [ 0, cos(x) - sin(x), 0],
>     [ 0, 0, 0 ] ] ):
> 'T(phi)'= T(phi);
```

$$T(\phi) = \begin{bmatrix} \cos(\phi) + \sin(\phi) & 0 & 0 \\ 0 & \cos(\phi) - \sin(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
> # define macros for greek characters
> macro( a=alpha, b=beta, c=gamma, f=phi, t=theta ):
> # coordinate mapping Y is a product of matrices
> Y := evalm( r/sqrt(2) * ( R[z](a) &* R[y](b) &* R[z](c/2)
>     &* T(t/2) &* R[z](f/2) ) ):
> # compute the metric tensor G
> Y1 := map( diff, Y, r ): Y2 := map( diff, Y, a ):
> Y3 := map( diff, Y, b ): Y4 := map( diff, Y, c ):
> Y5 := map( diff, Y, t ): Y6 := map( diff, Y, f ):
> # build the metric tensor
> G := array( symmetric, 1..6, 1..6 ):
> for i to 6 do for j from i to 6 do
> G[i,j] := simplify( trace( transpose(Y.i) &* Y.j ) )
> od od:
> intermediate_cpu_time = (time() - settime) * seconds;
```

$$intermediate_cpu_time = 6.120\ seconds$$

A [198]-ban található formulák eléréséhez most néhány egyszerűsítő eljárást alkalmazunk. A szekció outputjának lerövidítése végett továbbra is elnyomjuk a legtöbb részeredmény kiíratását. Amit látni fogunk, már egy „polírozott verzió”, nem a problémát megoldó legelső első interaktív szekció.

```
> G := subs( cos(t/2)^2 = 1/2 + 1/2*cos(t),
>     cos(c/2)^2 = 1/2 + 1/2*cos(c), sin(t/2) = sin(t) /
>     ( 2*cos(t/2) ), sin(c/2) = sin(c) / ( 2*cos(c/2) ),
>     eval(G)):
> G := map( normal, G ): # normalize each matrix entry
> G[2,2] := normal( subs( -1/2*r^2*cos(c) * sin(t) =
>     -1/2*r^2*(cos(b)^2 + sin(b)^2) * cos(c) * sin(t),
>     G[2,2] ) ):
> G := map( factor, G ): # this the formula in the paper!
> G[2,2] := subsop( 1=-op(1,G[2,2]), 3=-op(3,G[2,2]),
>     G[2,2] ): # get nice signs for entry 2,2
> print(G); # this the formula in the paper!
```

$$[1, 0, 0, 0, 0, 0]$$

$$\left[0,\ \frac{1}{2} r^2 (\cos(\beta)^2 + 1 - \sin(\theta)\cos(\gamma)\sin(\beta)^2),\right.$$

$$\left.\frac{1}{2} r^2 \sin(\gamma)\sin(\beta)\sin(\theta),\ \frac{1}{2} r^2 \cos(\beta),\ 0,\ \frac{1}{2} r^2 \cos(\beta)\cos(\theta)\right]$$

$$\left[0\,,\frac{1}{2}\,r^2\sin(\gamma)\sin(\beta)\sin(\theta)\,,\frac{1}{2}\,r^2\,(\sin(\theta)\cos(\gamma)+1)\,,0\,,0\,,0\right]$$

$$\left[0\,,\frac{1}{2}\,r^2\cos(\beta)\,,0\,,\frac{1}{4}\,r^2\,,0\,,\frac{1}{4}\,r^2\cos(\theta)\right]$$

$$\left[0\,,0\,,0\,,0\,,\frac{1}{4}\,r^2\,,0\right]$$

$$\left[0\,,\frac{1}{2}\,r^2\cos(\beta)\cos(\theta)\,,0\,,\frac{1}{4}\,r^2\cos(\theta)\,,0\,,\frac{1}{4}\,r^2\right]$$

A véges képernyőméret és a hosszú kifejezések miatt a Maple outputja nem túl szép, de a kapott kifejezések automatikusan lefordíthatók például a LaTeX szövegformázó rendszernek megfelelő formátumra. (V. ö. [124].)

```
> latex( G, 'metric_tensor' ):
```

Az outputot itt sem írattuk ki, a LaTeX által kiszedett végeredmény:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0\\
0 & \frac{r^2\left(\cos(\beta)^2+1-\sin(\theta)\cos(\gamma)\sin(\beta)^2\right)}{2} & \frac{r^2\sin(\gamma)\sin(\beta)\sin(\theta)}{2} & \frac{r^2\cos(\beta)}{2} & 0 & \frac{r^2\cos(\beta)\cos(\theta)}{2}\\
0 & \frac{r^2\sin(\gamma)\sin(\beta)\sin(\theta)}{2} & \frac{r^2(\sin(\theta)\cos(\gamma)+1)}{2} & 0 & 0 & 0\\
0 & \frac{r^2\cos(\beta)}{2} & 0 & \frac{r^2}{4} & 0 & \frac{r^2\cos(\theta)}{4}\\
0 & 0 & 0 & 0 & \frac{r^2}{4} & 0\\
0 & \frac{r^2\cos(\beta)\cos(\theta)}{2} & 0 & \frac{r^2\cos(\theta)}{4} & 0 & \frac{r^2}{4}
\end{bmatrix}$$

Számoljuk ki a Jacobi mátrix determinánsát.

```
> determinant := simplify( det( G ) ):
> determinant := normal( subs( cos(b)^2 = 1 - sin(b)^2,
>   cos(t)^4 = cos(t)^2 * (1 - sin(t)^2), determinant) ):
> assume( 0<=r, 0<=t, t<=Pi/2, 0<=b, b<=Pi ):
> Jacobian := subs( cos(t) = sin(2*t) / (2*sin(t)),
>   sqrt( determinant ) );
```

$$Jacobian := \frac{1}{32}\,r^{\sim 5}\sin(2\,\theta^\sim)\sin(\beta^\sim)$$

Ez a cikkben szereplő (33) formula. A hiperszférikus koordinátákra tett föltételekre amiatt volt szükség, hogy a Maple automatikusan elvégezze a négyzetgyökös kifejezések egyszerűsítését. A változónevek utáni tilde karakterek azt jelzik, hogy valamely föltételeket tettünk az illető változóra.

Határozzuk meg a metrikus tenzor inverzét:

```
> GINV := map( simplify, inverse( G ) ):
> GINV := subs( cos(t)^2 = 1 - sin(t)^2,
>   cos(b)^2 = 1 - sin(b)^2, eval( GINV ) ):
> cpu_time = (time()-settime) * seconds; # computing time
```

$$cpu_time = 7.509\,seconds$$

A metrikus tenzor inverzét nem írattuk ki, GINV[4,4] kivételével minden komponense a cikk (34) formulájának megfelelő alakú. A GINV[4,4]-et is könnyen a kívánt alakra hozhatnánk, de a számolás ezen utolsó fázisát inkább átugorjuk. Végül is néhány másodpercnyi gépidő fölhasználásával már publikálható eredményeket kaptunk.

A [45] cikkben egy másik olyan példát találunk, amikor a számítógépes algebra használatával tudtak a kutatók egy olyan bizonyítást befejezni, amely rengeteg triviális, de fáradságos számítást igényelt. Az említett cikkben egy tisztán algebrai problémát, nevezetesen véges Lie csoportok bizonyos véges részcsoportjainak létezését visszavezették egy 240 egyenletből álló lineáris egyenletrendszer megoldására, melynek együtthatói egy 1831 elemű testből kerültek ki. Számítógép segítségével könnyen meg tudták mutatni, hogy ennek a lineáris egyenletrendszernek van megoldása. Míg kézzel ez szinte lehetetlen, számítógéppel könnyedén kivitelezhető. A Maple segítségével végrehajtott számításokról lásd a [131]-et.

Utolsó két kidolgozott példánkban a Maple-lel a generátorfüggvényt, illetve a metrikus tenzort és inverzét határoztuk meg. Lehet, hogy az Olvasó nem túl sokra tartja a számítógépes programokkal kapott matematikai eredményeket, de ne feledje, hogy számos független módszer létezik a válaszok ellenőrzésére. Az egyik út az lehet, hogy magát a Maple-t használjuk a számítások ellenőrzésére, megnövelve ezzel az eredménybe vetett bizalmunkat. Így például kiszámíthatjuk a Taylor-sor első néhány elemét, és összehasonlíthatjuk az eredeti együtthatókkal, a válaszok további ellenőrzéseként pedig összeszorozhatjuk a metrikus tenzort inverzével.

A szimbolikus és algebrai számítások gyakran megelőzik a numerikus számításokat. A matematikai formulákat gyakran azért alakítjuk át, hogy a később elvégzendő numerikus számításoknak megfelelő alakra hozzuk őket. A számítógépes algebrai rendszerek fontos tulajdonsága, hogy jó kapcsolódási felületet kínálnak a számítások két típusa között. A Maple FORTRAN és C kifejezésekké tudja alakítani a saját nyelvén fölírt kifejezéseket. Opcionális dupla pontosságú aritmetikát és kódoptimalizálást is tud végezni. Például FORTRAN kóddá transzformálható az előző példánkban szereplő G metrikus tenzor első két sorából álló részmátrix is. Két technikai apróság: nem hivatkozhatunk az Euler-konstansra a γ szimbolummal, és a mátrixokat nem adhatjuk meg egymásba skatulyázott listákkal.

```
> `evalf/constant/gamma` := proc() args end:
> H := submatrix( G, 2..3, 1..6 );
```

$$H :=$$

$$\left[0\,,\, -\frac{1}{2}\, r^2\, (-\cos(\beta)^2 + \sin(\theta)\cos(\gamma)\sin(\beta)^2 - 1)\,, \right.$$

$$\left. \frac{1}{2}\, r^2 \sin(\gamma)\sin(\beta)\sin(\theta)\,,\, \frac{1}{2}\, r^2\cos(\beta)\,,\, 0\,,\, \frac{1}{2}\, r^2\cos(\beta)\cos(\theta) \right]$$

$$\left[0\,,\, \frac{1}{2}\, r^2 \sin(\gamma)\sin(\beta)\sin(\theta)\,,\, \frac{1}{2}\, r^2\, (\sin(\theta)\cos(\gamma) + 1)\,,\, 0\,,\, 0\,,\, 0 \right]$$

```
> fortran( H, 'optimized', precision = double );

t2 = sin(theta)
   t4 = t2*cos(gamma)
   t5 = sin(beta)
   t6 = t5**2
   t8 = cos(beta)
   t9 = t8**2
   t15 = t1*sin(gamma)*t5*t2
   t16 = t1*t8
   H(1,1) = 0.D0
   H(1,2) = t1*(-t4*t6+1.D0+t9)/2.D0
   H(1,3) = t15/2.D0
   H(1,4) = t16/2.D0
   H(1,5) = 0.D0
   H(1,6) = t16*cos(theta)/2.D0
   H(2,1) = 0.D0
   H(2,2) = t15/2.D0
   H(2,3) = t1*(t4+1.D0)/2.D0
   H(2,4) = 0.D0
   H(2,5) = 0.D0
   H(2,6) = 0.D0
```

A számítógépes algebrai rendszerekkel nyert eredmények vagy egzaktak, vagy a fölhasználó által meghatározott pontosságúak. A kézzel végzett számításoknál pontosabbak is lehetnek (v. ö. [116]); ami gyakran az integrál-táblázatok korrekciójához vezet. Az alábbiakban két olyan integrált említünk példaként, amelyek hibásan szerepelnek a legnépszerűbb táblázatokban. (V. ö. Gradshtein és Ryzhnik, [84].)

1. A 2.269 képlet

$$\int \frac{1}{x\sqrt{(bx+cx^2)^3}}\,dx = \frac{2}{3}\left(-\frac{1}{bx}+\frac{4c}{b^2}-\frac{8c^2x}{b^3}\right)\frac{1}{sqrt(bx+cx^2)}$$

2. A 3.828(19) képlet

$$\int_0^\infty \frac{\sin^2(ax)\sin^2(bx)\sin(2cx)}{x}\,dx =$$

$$\frac{\pi}{16}\left(1+\operatorname{sgn}(c-a+b)+\operatorname{sgn}(c-b+a)-2\operatorname{sgn}(c-a)-2\operatorname{sgn}(c-b)\right),$$

ahol $a, b, c > 0$.

Mint alább látható, a Maple nemcsak a helyes válaszokat adja meg, hanem a paraméterekre vonatkozóan további információkat is nyújt. A példában a " kettős idézőjellel hivatkozunk az előző eredményre.

```
> Int( 1 / (x * sqrt(( b*x + c*x^2 )^3)), x );
```

$$\int \frac{1}{x\,\sqrt{(b\,x+c\,x^2)^3}}\,dx$$

```
> value(");
```

$$\frac{2}{3}\sqrt{x\,(c\,x+b)}(4\sqrt{b\,x+c\,x^2}\,c\,x\,b+5\sqrt{b\,x+c\,x^2}\,c^2\,x^2 - \sqrt{b\,x+c\,x^2}\,b^2+3\,c^2\,\sqrt{x\,(c\,x+b)}\,x^2)\,/(x\,\sqrt{x^3\,(c\,x+b)^3}\,b^3)$$

```
> factor(");
```

$$\frac{2}{3}\,\frac{(c\,x+b)\,(4\,c\,x\,b+8\,c^2\,x^2-b^2)}{\sqrt{x^3\,(c\,x+b)^3}\,b^3}$$

Tegyük föl, hogy minden változó pozitív, és egyszerűsítsük az előző eredményt ennek megfelelően.

```
> simplify( ", assume=positive );
```

$$\frac{2}{3}\,\frac{4\,c\,x\,b+8\,c^2\,x^2-b^2}{x^{3/2}\,\sqrt{c\,x+b}\,b^3}$$

Kicsit több munkával jár a táblázatban szereplő alakra hozni az eredményt:

```
> subs( b+c*x = y/x, " );
```

$$\frac{2}{3}\,\frac{4\,c\,x\,b+8\,c^2\,x^2-b^2}{x^{3/2}\,\sqrt{\frac{y}{x}}\,b^3}$$

```
> simplify( ", assume=positive );
```

$$\frac{2}{3}\,\frac{4\,c\,x\,b+8\,c^2\,x^2-b^2}{x\,\sqrt{y}\,b^3}$$

```
> expand(");
```

$$\frac{8}{3}\,\frac{c}{\sqrt{y}\,b^2}+\frac{16}{3}\,\frac{x\,c^2}{\sqrt{y}\,b^3}-\frac{2}{3}\,\frac{1}{x\,\sqrt{y}\,b}$$

```
> collect( 3/2*", sqrt(y) );
```

$$\frac{4\,\frac{c}{b^2}+8\,\frac{x\,c^2}{b^3}-\frac{1}{x\,b}}{\sqrt{y}}$$

```
> subs( y = b*x+c*x^2, 2/3*" );
```

$$\frac{2}{3}\,\frac{4\,\frac{c}{b^2}+8\,\frac{x\,c^2}{b^3}-\frac{1}{x\,b}}{\sqrt{b\,x+c\,x^2}}$$

A táblázat tehát egy hibás – előjelet tartalmaz. Az 1980-as negyedik kiadásban egyébként már kijavítva szerepel. Különben is a táblázatokban közölt eredményeket az ember vagy elhiszi, vagy nem; a Maple-ben viszont differenciálással ellenőrizhetjük őket:

```
> simplify( diff(",x) - 1 / (x * sqrt(( b*x + c*x^2 )^3)),
>   assume=positive );
```

$$0$$

A második példa már izgalmasabb: a táblázatban szereplő hibás eredmény helyett a Maple egy általánosabb érvényű megoldást talál.

```
> Int( sin(a*x)^2 * sin(b*x)^2 *sin(2*c*x) / x,
>     x=0..infinity );
```

$$\int_0^\infty \frac{\sin(a\,x)^2 \sin(b\,x)^2 \sin(2\,c\,x)}{x}\,dx$$

```
> value(");
```

$$\frac{1}{32}\operatorname{signum}(2\,b+2\,c-2\,a)\,\pi-\frac{1}{16}\operatorname{signum}(2\,c-2\,b)\,\pi \\ +\frac{1}{32}\operatorname{signum}(-2\,b+2\,c-2\,a)\,\pi+\frac{1}{32}\operatorname{signum}(-2\,b+2\,c+2\,a)\,\pi \\ -\frac{1}{16}\operatorname{signum}(2\,c-2\,a)\,\pi+\frac{1}{8}\operatorname{signum}(c)\,\pi \\ +\frac{1}{32}\operatorname{signum}(2\,b+2\,c+2\,a)\,\pi-\frac{1}{16}\operatorname{signum}(2\,c+2\,a)\,\pi \\ -\frac{1}{16}\operatorname{signum}(2\,c+2\,b)\,\pi$$

A Maple tetszőleges a, b és c-re kiszámította a megoldást. Ezt specializálhatjuk a [84]-ben szereplő, csak a pozitív változóknak megfelelő esetre:

```
> simplify( ", assume=positive ); # all variables > 0
```

$$\frac{1}{32}\operatorname{signum}(2\,b+2\,c-2\,a)\,\pi-\frac{1}{16}\operatorname{signum}(2\,c-2\,b)\,\pi \\ +\frac{1}{32}\operatorname{signum}(-2\,b+2\,c-2\,a)\,\pi+\frac{1}{32}\operatorname{signum}(-2\,b+2\,c+2\,a)\,\pi \\ -\frac{1}{16}\operatorname{signum}(2\,c-2\,a)\,\pi+\frac{1}{32}\,\pi$$

Szabaduljunk meg a kettesektől, és állítsuk elő szorzatként az eredményt:

```
> factor( subs( signum = (signum @ primpart ), " ) );
```

$$\frac{1}{32}\pi(\operatorname{signum}(b+c-a)-2\operatorname{signum}(c-b)+\operatorname{signum}(-b+c-a) \\ +\operatorname{signum}(-b+c+a)-2\operatorname{signum}(c-a)+1)$$

A táblázatból tehát hiányzott egy tag, és rossz volt az egyik előjel.

Az integrálásnál sokszor csak akkor kaphatunk helyes eredményeket, ha megfelelő kikötéseket teszünk a paraméterekre. Vizsgáljuk az alábbi improprius integrált:

$$\int_0^\infty \frac{t^{1/3}\ln(at)}{(b+2t^2)^2}\,dt, \qquad a,b>0.$$

```
> assume( a>0, b>0 ):
> normal( integrate( t^(1/3) * ln(a*t) / (b + 2*t^2)^2,
> t = 0..infinity ) );
```

$$\frac{1}{36}\,\frac{\pi\,2^{1/3}\,(2\ln(a\tilde{}\,)\sqrt{3}+\pi-3\sqrt{3}-\sqrt{3}\ln(2)+\sqrt{3}\ln(b\tilde{}\,))}{b\tilde{}^{\,4/3}}$$

Az a és b utáni tildék azt jelzik, hogy ezekre a változókra valamilyen föltételek vannak érvényben. Szükség esetén a Maple tájékoztat is ezekről a föltételekről.

```
> about( a );

Originally a, renamed a~:
  is assumed to be: RealRange(Open(0),infinity)
```

Az integrálás jól illusztrálja a számítógépes algebra további előnyeit: könnyen elérhetővé válnak a bonyolultabb matematikai technikák és algoritmusok is. Ha a Maple-lel kiszámíttatjuk a következő két integrált

$$\int x^2 \exp(x^3)\, dx \quad \text{és} \quad \int x \exp(x^3)\, dx,$$

különböző jellegű eredményeket kapunk:

$$\frac{1}{3}\exp(x^3) \quad \text{és} \quad \int x \exp(x^3)\, dx.$$

Ez nem csak azt jelenti, hogy a rendszer nem tudott megbirkózni a második integrállal. A Maple a Risch algoritmus (lásd [76]) segítségével azt is el tudja dönteni, hogy az integrál nem fejezhető ki zárt alakban az elemi függvények segítségével. Ez ellentétes a zárt formájú megoldást kereső, szokásos heurisztikus algoritmusokkal. A heurisztikus megközelítések esetében soha nem lehetünk bizonyosak abban, hogy tényleg nem létezik zárt forma. Előfordulhat, hogy csupán nem elég ügyesen próbálkoztunk. A Risch algoritmus azonban igen bonyolult; mély matematikai eredményekre és algoritmusokra támaszkodik, és sok olyan lépésből áll, amelyek nehezen reprodukálhatók kézi számolással. A számítógépes algebrai rendszerek fölhasználói anélkül alkalmazhatják ezeket a mély matematikai eredményeket és algoritmusokat, hogy minden részletet ismerniük kellene.

Ha számítógépes algebrai rendszert használunk, a matematikai feladat elemzésére koncentrálhatunk, a számolás részleteit a gépre bízhatjuk. A számítógépes algebrai rendszerek „matematikai kísérletekre" buzdítanak bennünket. Megkönnyítik a matematikai hipotézisek ellenőrzését, és számitásainkon alapuló újabb sejtésekhez vezetnek. (V. ö. [15, 130].) Az ilyen matematikai kísérletek példája lehet az alábbi $n \times n$-es $A_n$mátrix determinánsára vonatkozó sejtésünk megfogalmazása:

$$A_n(i,j) = x^{\gcd(i,j)}.$$

Az első lépés, hogy kiszámolunk néhány determinánst, és megvizsgáljuk őket.

```
> numex := 8:  # number of experiments
> dets := array( 1..numex ):
> for n to numex do
>   A[n] := array( 1..n, 1..n, symmetric ):
>   for i to n do
>     for j to i do
>       A[n][i,j] := x^igcd(i,j)
>   od od:
```

```
>   dets[n] := factor( linalg[det]( A[n] ) ):
>   print( dets[n] )
> od:
```

$$x$$
$$x^2(x-1)$$
$$x^3(x+1)(x-1)^2$$
$$x^5(x+1)^2(x-1)^3$$
$$x^6(x^2+1)(x+1)^3(x-1)^4$$
$$x^7(x^2+1)(x^3+x-1)(x+1)^4(x-1)^5$$
$$x^8(x^2+1)(x^2+x+1)(x^2-x+1)(x^3+x-1)(x+1)^5(x-1)^6$$
$$x^{12}(x^2+x+1)(x^2-x+1)(x^3+x-1)(x^2+1)^2(x+1)^6(x-1)^7$$

Első látásra nem sok jóval biztat. De nézzük csak az egymás utáni determinánsok hányadosát!

```
> for i from 2 to numex do
>   quo( dets[i], dets[i-1], x )
> od;
```

$$x^2-x$$
$$x^3-x$$
$$x^4-x^2$$
$$x^5-x$$
$$x^6-x^3-x^2+x$$
$$x^7-x$$
$$x^8-x^4$$

Az n-dik polinomban csak olyan x hatványok fordulnak elő, amelyek kitevője n osztója. Ha még egy kicsit kísérletezünk a Maple-lel, és egy Möbius-szerű formulával próbálkozunk, könnyen eljuthatunk a következő sejtéshez:

Sejtés. $\det A_n = \prod_{j=1}^{n} \phi_j(x)$, *ahol a* $\phi_j(x)$ *polinomokat az* $x^n = \sum_{d|n} \phi_d(x)$ *egyenlőség definiálja.*

Néhány tulajdonság:

(i) Ha p prím, akkor

$$\phi_p(x) = x^p - x$$

és tetszőleges r természetes számra

$$\phi_{p^r}(x) = \phi_p(x^{p^{r-1}}).$$

(ii) Ha a p prím nem osztója n-nek, akkor

$$\phi_{pn}(x) = \phi_n(x^p) - \phi_n(x).$$

(*iii*) *Ha az n természetes szám prímtényezős fölbontása $n = p_1^{r_1} \ldots p_s^{r_s}$, akkor*

$$\phi_n(x) = \phi_{p_1 \ldots p_s}\left(x^{p_1^{r_1-1} \ldots p_s^{r_s-1}}\right).$$

(*iv*) *A következő formula érvényes:*

$$\phi_n(x) = \sum_{d|n} \mu\left(\frac{n}{d}\right) x^d = \sum_{d|n} \mu(d) x^{\left(\frac{n}{d}\right)}$$

ahol μ a Möbius-függvény, azaz $\mu(1) = 1$, $\mu(p_1 \ldots p_s) = (-1)^s$ ha $p_1, \ldots p_s$ különböző prímek és $\mu(m) = 0$, ha m valamely prímszám négyzetével osztható.

Az érdeklődő Olvasók kedvéért megjegyezzük, hogy $\frac{1}{d}\phi_d(q)$ a q elemű véges test fölötti egyváltozós d-ed fokú monikus irreducibilis polinomok számával egyenlő.

Bármilyen sok matematikai ismeretet tartalmaznak is a számítógépes algebrai rendszerek, fontos, hogy a gyakorlott fölhasználók személyes érdeklődésüknek megfelelő eljárások megírásával is bővíthessék a rendszert. A könyv szerzője az [59]-ben kifejlesztett módszerek alapján polinomiális leképezések invertálására néhány Maple algoritmust implementált. Ezekkel a programokkal eldönthető, hogy egy polinomiális leképezésnek van-e polinomiális inverze, és ha igen, ki is számolható az inverz. Ezt alátámasztandó, definiáljuk a következő háromváltozós invertálható leképezést:

```
> read 'invpol.m'; # load user-defined package
> P := [ x^4 + 2*(y+z)*x^3 + (y+z)^2*x^2 + (y+1)*x
>     + y^2 + y*z, x^3 + (y+z)*x^2 + y, x + y + z ];
```

$$P := [x^4 + 2\,(y+z)\,x^3 + (y+z)^2\,x^2 + (y+1)\,x + y^2 + y\,z,\\ x^3 + (y+z)\,x^2 + y,\, x + y + z]$$

```
> settime := time(): # start timing
> invpol( P, [x,y,z] ); # compute inverse mapping
```

$$[x - y\,z,\, y - x^2\,z + 2\,y\,z^2\,x - y^2\,z^3,\\ z - x - y + y\,z + x^2\,z - 2\,y\,z^2\,x + y^2\,z^3]$$

```
> cpu_time = (time()-settime) * seconds; # computing time
```

$$cpu_time = .430\,seconds$$

A polinomiális leképezések inverzét papírral és ceruzával szinte lehetetlen kiszámolni. Számítógépes algebrai rendszerre van szükség a rengeteg szimbolikus számításhoz. Azt azonban nem várhatjuk el a rendszerek tervezőitől, hogy az összes fölhasználó minden jövőbeli igényét figyelembe vegyék. Csak arra számíthatunk, hogy olyan hatékony programozási eszközökkel látnak el bennünket, amelyekkel új algoritmusokat implementálhatunk.

1.5. A számítógépes algebra korlátai

A számítógépes algebrai rendszerekről eddig elmondottak azt a benyomást kelthetik, hogy ezek a rendszerek korlátlan lehetőségeket kínálnak, és univerzális eszközként használhatók matematikai problémák megoldására. Ezek azonban túl rózsás elképzelések, nem árt néhány előzetes figyelmeztető megjegyzés.

A számítógépes algebrai rendszerek gyakran komolyan megterhelik az őket futtató számítógépeket, mivel rengeteg memóriát és számítási időt igényelnek. A pontos aritmetika ára a kifejezések méretének exponenciális növekedése és óriási nagy számok megjelenése. Ez még akkor is előfordulhat, ha a végső válasz egyszerű „igen" vagy „nem". Vegyük a jólismert euklideszi algoritmust, amely két polinom legnagyobb közös osztóját adja meg. A „naív" algoritmus hatékonysága nem túl jó. Tekintsük a következő polinomokat

```
> f[1] := 7*x^7 + 2*x^6 - 3*x^5 - 3*x^3 + x + 5;
```

$$f_1 := 7\,x^7 + 2\,x^6 - 3\,x^5 - 3\,x^3 + x + 5$$

```
> f[2] := 9*x^5 - 3*x^4 - 4*x^2 + 7*x + 7;
```

$$f_2 := 9\,x^5 - 3\,x^4 - 4\,x^2 + 7\,x + 7$$

Az euklideszi algoritmussal megkeressük a két polinom racionális számok fölötti legnagyobb közös osztóját, $\gcd(f_1, f_2)$-t. Az első osztás elvégzésével olyan q_2 és f_3 polinomokat kapunk, amelyekre $f_1 = q_2 f_2 + f_3$ teljesül, továbbá vagy $\mathrm{degree}(f_3) < \mathrm{degree}(f_2)$ vagy $f_3 = 0$. Ezt a lépést a következő maradékos osztással végezzük:

```
> f[3] := sort( rem( f[1], f[2], x, q[2] ) );
```

$$f_3 := \frac{70}{27}\,x^4 - \frac{176}{27}\,x^3 - \frac{770}{81}\,x^2 - \frac{94}{81}\,x + \frac{503}{81}$$

```
> q[2];
```

$$\frac{7}{9}\,x^2 + \frac{13}{27}\,x - \frac{14}{81}$$

Ezután olyan q_3 és f_4 polinomokat határozunk meg, amelyekkel $f_2 = q_3 f_3 + f_4$ teljesül, és vagy $\mathrm{degree}(f_4) < \mathrm{degree}(f_3)$ vagy $f_4 = 0$. Az osztásokat addig ismételjük, míg $f_n = 0$ nem lesz, ekkor $\gcd(f_1, f_2) = f_{n-1}$. A következő Maple program a maradékpolinomok sorozatát számítja ki:

```
> Euclid_gcd := proc( f::polynom, g::polynom, x::name )
>   local r:
>   if g = 0 then sort( f )
>   else
>     r := sort( rem( f, g, x ) );
>     if r <> 0 then print( r ) fi;
>       Euclid_gcd( g, r, x )
>   fi
> end:
> Euclid_gcd( f[1], f[2], x ):
```

$$\frac{70}{27}\,x^4 - \frac{176}{27}\,x^3 - \frac{770}{81}\,x^2 - \frac{94}{81}\,x + \frac{503}{81}$$

$$\frac{100881}{1225}x^3 + 72\,x^2 - \frac{14139}{2450}x - \frac{98037}{2450}$$
$$-\frac{16726864175}{10176976161}x^2 - \frac{5255280625}{10176976161}x + \frac{19754564375}{10176976161}$$
$$\frac{35171085032244648729}{4567967104145280 50}x + \frac{66056048950873335357}{45679671041452805 0}$$
$$\frac{2406814310427212456610119019 25}{1215493878315063455640258624 81}$$

Azt az eredményt kaptuk, hogy f_1 és f_2 relatív prímek, mivel legnagyobb közös osztójuk egység. Figyeljük meg az együtthatók méretének óriási növekedését az egyjegyű egészektől egészen a harmincjegyű racionális számokig (még úgy is, hogy a racionális együtthatók mindig egyszerűsített alakban szerepelnek). A [76, 117]-ben olvashatunk kifinomultabb algoritmusokról, melyek a legnagyobb közös osztót úgy számítják ki, hogy közben a lehetőségekhez mérten elkerülik az együtthatók ilyen mértékű növekedését.

A számítógépes algebrai számítások során gyakran előforduló jelenség, hogy a részszámításokban szereplő kifejezések mérete rendkívüli módon megnő, ez az „intermediate expression swell" (kb. „részkifejezések felfúvódása"). Bár általában nehéz megbecsülni a számítások idő- és memóriaigényét, mindent meg kell tennünk annak érdekében, hogy számításainkat alkalmas matematikai modellek használatával és hatékony programozással optimalizáljuk. Annak bizonyítására, hogy megéri a fáradságot, tekintsük azt a 8×8-as mátrixot, melynek (i,j)-dik komponense $(iu + x + y + z)^j$. Ennek inverzét a Maple 1125 másodperc alatt 3900 Kb memória fölhasználásával számolta ki egy SUN SPARCstation 20 M71 típusú munkaállomáson. A nyilvánvaló $x + y + z \to v$ helyettesítés a számítási időt 6 másodpercre, a memóriafölhasználást 750 Kb-ra csökkentette.

A számítógépes algebrai rendszerek használatával kapcsolatos második probléma pszichológiai jellegű. Hány soros output-ot lehet könnyedén áttekinteni? Ha nagy méretű kifejezésekkel találjuk magunkat szemben, hogyan lesz elegendő rálátásunk az egyszerűsítésükhöz? Ha például csak a kifejtett alakját látjuk a képernyőn, nehéz lenne azt az

$$f(x,y) = g(u(x,y), v(x,y)),$$

polimon-kompozíciót fölfedezni, amelyben

$$\begin{aligned} g(u,v) &= u^3v + uv^2 + uv + 5, \\ u(x,y) &= x^3y + xy^2 + x^2 + y + 1, \\ v(x,y) &= y^3 + x^2y + x. \end{aligned}$$

Bár igaz, hogy a számítógépes algebrai rendszerekben tetszőleges pontosságú numerikus számításokat végezhetünk, ennek negatív oldala is van. Mivel a hardver által kínált aritmetika helyett szoftveres lebegőpontos aritmetikát használunk, az így végrehajtott numerikus számítások 100 – 1000-szer lassúbbak például a

FORTRAN programozási nyelven megírtaknál. Numerikus problémák esetén ezért mindig föl kell tenni magunknak a kérdést: „Valóban szükségünk van-e a racionális számokkal dolgozó egzakt vagy a nagy pontosságú lebegőpontos aritmetikára? Nem lenne jobb valamelyik numerikus programozási nyelvet használni?"

A számítógépes algebrai rendszereket enyhe túlzással szokás matematikai szakértői rendszereknek is nevezni. Bármennyire impresszív is a rendszerekbe épített matematikai tudásanyag, ez csak napjaink matematikai ismereteinek tört része. Számos olyan területe van a matematikának, ahol a számítógépes algebra még nem sokat tud segíteni, és ahol további kutatások szükségesek. Ilyen területek a parciális differenciálegyenletek megoldása, a Bessel függvényekhez hasonló nem elemi függvényeket tartalmazó határozott és határozatlan integrálok kiszámítása, a vonalmenti és a felületi integrálás, a speciális függvények kalkulusa és a nem kommutatív algebra, hogy csak néhányat említsünk.

További súlyos probléma, s talán ez a legnehezebb, ha absztrakt szinten akarjuk specifikálni azt a számtestet, amelyben számolni kívánunk. Például végtelen sok számtest létezik, de a testek aritmetikai műveleteinek tulajdonságait fölhasználó olyan algoritmusokat szeretnénk írni, ahol nem kell előre megadni azt a testet, amelyben a számításokat végezzük. Továbbá elképzelhető, hogy az Olvasó saját maga által definiált matematikai struktúrákat akar használni. A Scratchpad II/AXIOM [54, 55, 108, 109, 179] rendszer tette meg az első lépéseket ebben az irányban. A Maple-ben a `Gauss` csomaggal hozhatunk létre az AXIOM-hoz hasonló módon új doméneket.

Ami a számítógépes algebrai rendszerek szintaxisát és a szemantikáját illeti, programozási nyelvként való használatuk bonyolultabb, mint a FORTRAN-szerű numerikus programozási nyelveké. A számítógépes algebrai rendszerekben számos olyan beépített függvénnyel találkozunk, amelyek váratlan eredményekre vezethetnek. Valamiféle elképzelésünknek kell lenni a rendszer működéséről, a belső adatábrázolásról, az adatok gazdaságos kezeléséről stb. A matematikai problémák alapos vizsgálatára és az algoritmusok gondos implementálására van szükség, hogy mind az időigény, mind a memória-használat szempontjából hatékony algoritmusokat kapjunk. Ha például a Bessel polinomok kiszámítását végző eljárásban (lásd az 1.3. alfejezetet) elfelejtettük volna az **expand** eljárás segítségével kifejteni a részeredményeket, szükségtelenül terjedelmes kifejezéseket kaptunk volna. Az **Y** függvény eléggé szerencsétlen alábbi implementációja $Y_5(z)$-re ezt adja:

```
> Y := proc( n::nonnegint, x::name )
>   if n=0 then 1
>   elif n = 1 then x+1
>   else Y(n,x) := (2*n-1)*x*Y(n-1,x) + Y(n-2,x)
>   fi
> end:
> Y(5,z);
```

$$9\,z\,(7\,z\,(5\,z\,(3\,z\,(z+1)+1)+z+1)+3\,z\,(z+1)+1)$$
$$+\,5\,z\,(3\,z\,(z+1)+1)+z+1$$

A hatékony programozást, valamint a szimbolikus számításoknál gyakran előforduló számos csapda elkerülését megkönnyíti a számítógépes algebrai rendszer alapvető tulajdonságaiban való jártasság, például az elemi adatstruktúrák és a beépített lehetőségek ismerete. Valamely probléma vizsgálata során a rendelkezésünkre álló rendszerek alapos megértése a helyes rendszer kiválasztásának is előfeltétele. Például a FORM [184] formula-manipulációs rendszer nem kommutatív algebrákban való számítások elvégzésére alkalmasabb a Maple-nél, mivel a nem kommutatív objektumokat alaptípusként alkalmazza, és a mintaillesztés szintén megfelelően támogatott alapművelet. A [46]-ban kvaterniókkal végzett számításokat hasonlítanak össze a két rendszerben.

Végül megemlítjük a számítógépes algebrai rendszerek használatának technikai nehézségeit. Néha zavaró az input és az output formátuma, a kifejezések ismételt fölhasználása nehézkes, és túl messze esik a szokásos matematikai jelölésektől. A globális manipulációkhoz képest a részformulák lokális átalakításai számos esetben nehezen kivitelezhetőek. Más programozási nyelvekhez való kapcsolódás gyakran teljesen hiányzik, vagy nem megfelelő. És végül, de nem utolsó sorban a hibákról sem feledkezhetünk meg. Az érdeklődő Olvasónak a [175]-ben található szórakoztató diszkussziót javasoljuk. Remélhetjük, hogy az ottani megjegyzések közül néhány hamarosan idejétmúlttá válik. De a Maple-höz hasonló rendszerek mindig meglepetésekkel szolgálnak; hol jó, hol rossz értelemben. Ezt a részt két olyan meglepő példával zárjuk, ahol a Maple kis segítségre szorul.

Az első feladat egy olyan könnyű integrálás, amely számítógépes algebrai rendszer használata nélkül is megoldható:

```
> Int( 2*x*(x^2+1)^24, x );
```

$$\int 2\,x\,(x^2+1)^{24}\,dx$$

A válasz világos, de mit mond a Maple?

```
> value(");
```

$$19228\,x^{36}+43263\,x^{34}+81719\,x^{32}+\frac{653752}{5}\,x^{30}+178296\,x^{28}$$
$$+208012\,x^{26}+208012\,x^{24}+178296\,x^{22}+7084\,x^{38}+\frac{10626}{5}\,x^{40}$$
$$+\frac{1}{25}\,x^{50}+x^{48}+12\,x^{46}+92\,x^{44}+506\,x^{42}+\frac{10626}{5}\,x^{10}+506\,x^{8}$$
$$+\frac{653752}{5}\,x^{20}+81719\,x^{18}+43263\,x^{16}+19228\,x^{14}+7084\,x^{12}$$
$$+92\,x^{6}+12\,x^{4}+x^{2}$$

Hát ez eléggé meglepő! Talán a szorzattá alakítás segít.

```
> factor(");
```

$$\frac{1}{25}x^2\,(x^8+5\,x^6+10\,x^4+10\,x^2+5)(x^{40}+20\,x^{38}+190\,x^{36}$$
$$+1140\,x^{34}+4845\,x^{32}+15505\,x^{30}+38775\,x^{28}+77625\,x^{26}$$

$$+ 126425\,x^{24} + 169325\,x^{22} + 187760\,x^{20} + 172975\,x^{18}$$
$$+ 132450\,x^{16} + 84075\,x^{14} + 43975\,x^{12} + 18760\,x^{10} + 6425\,x^{8}$$
$$+ 1725\,x^{6} + 350\,x^{4} + 50\,x^{2} + 5)$$

Közben rájöhettünk, hogy az integrációs konstans választása okozta a gondokat:

```
> factor( " + 1/25 );
```

$$\frac{1}{25}(x^2+1)^{25}$$

Túl sokat várunk el a rendszertől, ha azt hisszük, hogy az efféle problémák mindig elkerülhetők.

A második példa optikai vizsgálatokból származik (lásd [14]), és

$$x^n \sin^i x \cos^j x \cosh^k x \sinh^l x$$

alakú tagokból álló kifejezések integrálásával kapcsolatos ($i, j, k, l \in \mathbb{N}$). Könnyen igazolható, hogy minden ilyen integrál kifejezhető hasonló fölépítésű tagok segítségével. Lássuk, mit csinál a Maple:

```
> Int( x * sin(x)^2 * cos(x) * sinh(x)^2 * cosh(x), x );
```

$$\int x \sin(x)^2 \cos(x) \sinh(x)^2 \cosh(x)\, dx$$

```
> value(");
```

$$\frac{1}{32}\left(\frac{3}{10}x - \frac{2}{25}\right)e^{(3\,x)}\cos(x) - \frac{1}{32}\left(-\frac{1}{10}x + \frac{3}{50}\right)e^{(3\,x)}\sin(x)$$
$$- \frac{1}{192}\,x\,e^{(3\,x)}\cos(3\,x) + \frac{1}{32}\left(-\frac{1}{6}x + \frac{1}{18}\right)e^{(3\,x)}\sin(3\,x)$$
$$- \frac{1}{64}\,x\,e^{x}\cos(x) + \frac{1}{32}\left(-\frac{1}{2}x + \frac{1}{2}\right)e^{x}\sin(x)$$
$$+ \frac{1}{32}\left(\frac{1}{10}x + \frac{2}{25}\right)e^{x}\cos(3\,x) - \frac{1}{32}\left(-\frac{3}{10}x + \frac{3}{50}\right)e^{x}\sin(3\,x)$$
$$+ \frac{1}{64}\,x\,e^{(-x)}\cos(x) + \frac{1}{32}\left(-\frac{1}{2}x - \frac{1}{2}\right)e^{(-x)}\sin(x)$$
$$+ \frac{1}{32}\left(-\frac{1}{10}x + \frac{2}{25}\right)e^{(-x)}\cos(3\,x) - \frac{1}{32}\left(-\frac{3}{10}x - \frac{3}{50}\right)e^{(-x)}\sin(3\,x)$$
$$+ \frac{1}{32}\left(-\frac{3}{10}x - \frac{2}{25}\right)e^{(-3\,x)}\cos(x) - \frac{1}{32}\left(\frac{1}{10}x - \frac{3}{50}\right)e^{(-3\,x)}\sin(x)$$
$$+ \frac{1}{192}\,x\,e^{(-3\,x)}\cos(3\,x) + \frac{1}{32}\left(-\frac{1}{6}x - \frac{1}{18}\right)e^{(-3\,x)}\sin(3\,x)$$

Ahhoz, hogy a kívánt alakú eredményt kapjuk, az exponenciális függvényeket a hiperbolikus függvényekkel kell kifejezni, és át kell alakítani az előző részeredményt:

```
> convert( ", 'trig' ): # exp -> trigonometric function
> expand(");
```

$$\frac{1}{15}\sin(x)\,x\cosh(x)^3-\frac{1}{10}\cos(x)\,x\sinh(x)$$
$$-\frac{1}{6}\,x\sinh(x)\cosh(x)^2\cos(x)^3-\frac{1}{50}\cos(x)\cosh(x)^3$$
$$+\frac{19}{450}\sin(x)\sinh(x)+\frac{1}{50}\cosh(x)\cos(x)^3$$
$$+\frac{1}{5}\cos(x)\,x\sinh(x)\cosh(x)^2-\frac{13}{450}\sin(x)\sinh(x)\cosh(x)^2$$
$$-\frac{13}{450}\sinh(x)\sin(x)\cos(x)^2-\frac{1}{6}\,x\cosh(x)^3\sin(x)\cos(x)^2$$
$$+\frac{1}{5}\,x\cosh(x)\sin(x)\cos(x)^2+\frac{1}{18}\sinh(x)\cosh(x)^2\sin(x)\cos(x)^2$$
$$-\frac{1}{10}\sin(x)\,x\cosh(x)+\frac{1}{15}\,x\sinh(x)\cos(x)^3$$

Utolsó lépésként ezeket a parancsokat egy eljárásban foglalhatnánk össze, hogy máskor is használhassuk őket:

```
> trigint := proc()
>   int(args);
>   convert(",'trig');
>   expand(")
> end:
> trigint( x*sin(x)^2*cos(x)^3, x );
```

$$\frac{1}{15}\,x\sin(x)\cos(x)^2+\frac{2}{15}\,x\sin(x)+\frac{1}{45}\cos(x)^3+\frac{2}{15}\cos(x)$$
$$-\frac{1}{5}\,x\cos(x)^4\sin(x)-\frac{1}{25}\cos(x)^5$$

```
> trigint( x*sin(x)*cos(x)^2, x=0..Pi );
```

$$\frac{1}{3}\pi$$

Gyakran előfordul, hogy a Maple-t így szabjuk hozzá igényeinkhez. Ezért a könyv számos példát mutat arra, hogyan segíthetünk a rendszernek. Némelyik eléggé bonyolult. Azért vettük föl őket, mert pusztán a Maple parancsok elmagyarázása és triviális példák fölsorolása még nem ad elegendő jártasságot ahhoz, hogy komolyan tudjunk dolgozni a rendszerrel.

1.6. A Maple tervezése

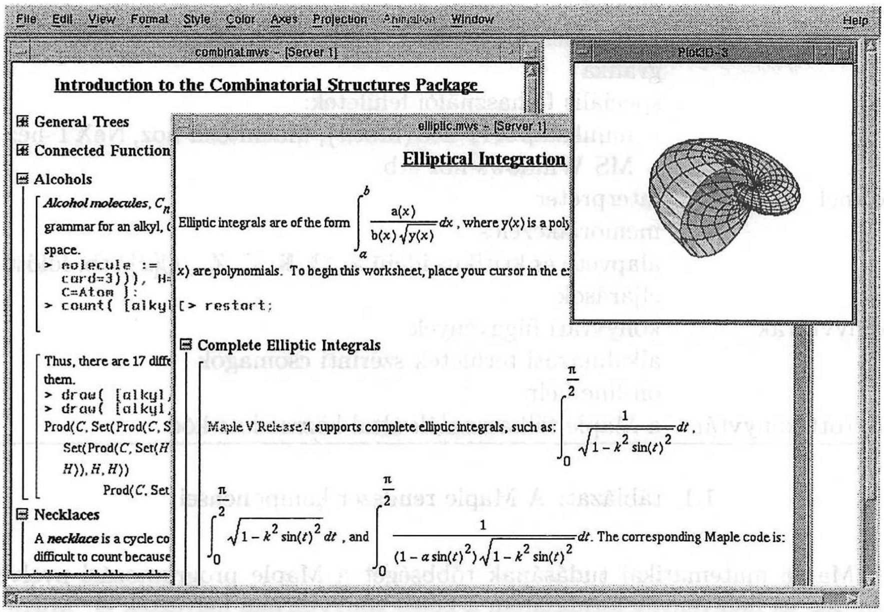
A Maple elnevezése nem a **ma**thematical **ple**asure („matematikai örömök") kifejezés rövidítéseként jött létre, csupán a termék kanadai származására utal.[5] 1980 novembere óta a Waterloo-i Egyetemen működő Szimbolikus Számítási Kutatócsoport és a Zürichi Műegyetem szakemberei óriási mennyiségű munkát öltek a Maple rendszer fejlesztésébe. 1992 óta a termék további fejlesztését és

[5] A maple (juharfa) Kanada nemzeti szimboluma. (*A Fordító megjegyzése.*)

forgalmazását az eredeti fejlesztőkkel együttműködve a Waterloo Maple Software (1995 óta Waterloo Maple Inc.) nevű cég végzi.

A Maple nyitott architectúrájú számítógépes algebrai rendszer, amely számítógépek széles skáláján futtatható a Cray Y/MP szuperkomputertől lefelé kezdve egészen a Macintosh és IBM PC kompatibilis asztali gépekig. A rendszer leghatékonyabban időosztásos operációs rendszert működtető nagygépeken futtatható, ahol számos fölhasználó dolgozhat egyidejűleg a Maple-lel anélkül, hogy egymást vagy más fölhasználókat zavarnának, vagy hogy a rendszert különösebben leterhelnék. Mindez a Maple moduláris tervezésének köszönhető. A rendszer négy részből áll: az *Iris*-nek nevezett fölhasználói felületből, az alapvető algebrai műveleteket végző *kernel*-ből, a külső *könyvtár*-ból és a Maple fölhasználói által fejlesztett *osztott könyvtár*-ból (share library).

Az Iris és a kernel a rendszer kisebbik részét alkotja. Mindkettőt a C programozási nyelven írták meg. Ezek töltődnek be a memóriába, amikor egy Maple szekciót elindítunk. Az Iris kezeli a matematikai kifejezések inputját (elemzés, hibaüzenetek), megjeleníti a kifejezéseket („prettyprinting”), kirajzolja a függvényeket, és a rendszernek a fölhasználóval folytatott egyéb kommunikációját támogatja. Az X Window rendszeren (Motif alatt), továbbá VMS, Amiga, Macintosh, NeXT és MS-Windows operációs rendszerű gépeken használható a *munkalapnak* (worksheet) nevezett speciális grafikus fölhasználói felület.



1.3. ábra: Maple ablak több munkalappal

A munkalapokon a Maple inputot és outputot grafikával és egyéb szöveggel kombinálhatjuk. Ennek tipikus példája látható az 1.3. ábrán. A Maple munkalapos felületének további sajátosságai: hipertext lehetőségeket biztosít a dokumentu-

mokon belül és különböző dokumentumok között, bizonyos platformokon megengedi multimédia objektumok beágyazását, nyomdai minőségben jeleníti meg a matematikai formulákat, az outputban szereplő képletek részei kiválaszthatók további földolgozás céljából. A Maple munkalap régiók hierarchiájából áll, a régiók fejezetekbe és alfejezetekbe foglalhatók össze. Ezek „becsukhatók”, ha tartalmukat el akarjuk rejteni. A munkalapok szöveges vagy LaTeX formátumban exportálhatók. A munkalapos interfészről pontosabb részleteket a [187, 188] Maple dokumentációban találhatunk.

A Maple kernel értelmezi a fölhasználói inputot, végrehajtja az alapvető algebrai műveleteket, mint például a racionális aritmetikát és az elemi polinomiális aritmetikát. Hatékonysági megfontolások alapján tartalmaz bizonyos gyakran használt algebrai rutinokat is. Ebbe a kategóriába tartoznak például a polinomok manipulációját végző **degree**, **coeff** és **expand** parancsok. Ugyancsak a kernel végzi a memóriakezelést. A Maple rendszer rendkívül fontos sajátossága, hogy a szekció alatt használt összes (rész)kifejezést csak egy-egy példányban tárolja. Ily módon a kifejezések egyenlőségének tesztelése rendkívül „olcsó” művelet, egyetlen gépi utasítással elvégezhető. A rendszer a részkifejezéseket újra felhasználja anélkül, hogy újra kiszámolná őket.

Részegység	**Funkció**
Iris	elemző
	kifejezések megjelenítése (prettyprinting)
	grafika
	speciális fölhasználói felületek:
	munkalap X11-hez (Motif), Macintosh-hoz, NeXT-hez, MS Windows-hoz stb.
Kernel	interpreter
	memóriakezelés
	alapvető és kritikus idejű $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}_n$, $\mathbb{Q}[x]$ stb. fölötti eljárások
Könyvtárak	könyvtári függvények
	alkalmazási területek szerinti csomagok
	on-line help
Osztott könyvtár	a Maple fölhasználók által közreadott kód

1.1. táblázat: A Maple rendszer komponensei

A Maple matematikai tudásának többségét a Maple programozási nyelvén kódolták, és függvények formájában a külső könyvtárban helyezték el. Mikor a felhasználónak szüksége van egy könyvtári függvényre, a Maple az esetek többségében van olyan okos, hogy magától betölti a rutint a memóriába. Csak a ritkán használatos Maple eljárások betöltésére kell explicit módon megkérni a rendszert. A Maple-t arról is tájékoztatnunk kell, ha valamelyik különálló csomagot kívánjuk elérni (ilyenek a lineáris algebra, a számelmélet és a statisztikai

csomag, csak hogy néhányat megnevezzünk közülük). Mindez a Maple-t kompakt, könnyen felhasználható rendszerré teszi. Ami még fontosabb, így a Maple csak lényeges dolgokat tárol a memóriában, semmi olyat nem, amire a fölhasználó nem kíváncsi. Ezért használhatják egy gépen egyidejűleg többen is a rendszert, és ezért fut a Maple kevés memóriával ellátott gépeken is. Az 1.1. táblázat összefoglalja a Maple rendszer korábban tárgyalt tervezési alapelveit.

A Maple nyelve jól strukturált, áttekinthető, magas szintű programozási nyelv. Adatstruktúrák széles választékát támogatja: függvényeket, sorozatokat, halmazokat, listákat, táblákat, stb. Számos, ezeken az adatstruktúrákon definiált, könnyen használható művelet is rendelkezésünkre áll, mint például a típusellenőrzés, szelekció, adatstruktúrák kompozíciója stb. Ezek a lényeges összetevői annak a programozási nyelvnek, amelyben a Maple majdnem minden matematikai algoritmusát implementálták, és amelyet mi is használunk, amikor a rendszert „interaktív számológépként" alkalmazzuk. Továbbá ha valakit érdekelnek a Maple által használt algoritmusok és ezek implementálása, megnézheti a könyvtárakban található Maple kódot. A könyvtári eljárások ugyanis forráskódban rendelkezésre állnak. Szükség esetén ezek a könyvtárak a fölhasználók saját fejlesztésű eljárásaival és csomagjaival is bővíthetők. A Maple osztott könyvtára, amely a Waterloo Maple Inc. WWW szerverén, a `http://www.maplesoft.com` URL-en érhető el, tartalmaz többek között egy csomó fölhasználók által közreadott kódot, munkalappéldákat, dokumentációt és a Maple korábbi verzióiban található hibák javításait. A Maple biztosít olyan lehetőségeket is, amelyekkel a programok végrehajtását nyomon követhetjük, függetlenül attól, hogy saját magunk által írt eljárásokról van-e szó.

A Maple előnyei közül utolsóként megemlítjük a rendszer fölhasználóbarát tervezését. Több számítógépes algebrai rendszernél hosszú tanulóidőre vagy alacsony szintű programozási nyelvi ismeretekre van szükség, ha valaki valóban meg akarja érteni, hogy mi is történik számítások során. Sok rendszerben vaskos kézikönyveket kell átlapozni, hogy megtaláljuk a programozási kapcsolók és kulcszavak helyes beállításait. A Maple-ben nincs erre szükség. Először is használhatjuk a help lehetőségeit, ezzel alapvetően a Maple eljárások on-line kézikönyvét kapjuk. Másodszor, a Maple könnyű használhatóságának titka az a hibrid algoritmikus struktúra, ahol a rendszer maga el tudja dönteni, hogy melyik algoritmus végrehajtása az előnyös. Példaként vessünk egy pillantást a Maple **simplify** („egyszerűsít") eljárására, ami pontosan azt csinálja, amit a neve sugall.

```
> trig_formula := cos(x)^6 + sin(x)^6
>   + 3*sin(x)^2*cos(x)^2:
> exp_ln_formula := exp( a +1/2*ln(b) ):
> radical_formula := (x-2)^(3/2) / (x^2-4*x+4)^(1/4):
> trig_formula = simplify( trig_formula );
```

$$\cos(x)^6 + \sin(x)^6 + 3\sin(x)^2\cos(x)^2 = 1$$

```
> exp_ln_formula = simplify( exp_ln_formula );
```

$$e^{(a+1/2\ln(b))} = e^a\sqrt{b}$$

```
> radical_formula = simplify( radical_formula );
```

$$\frac{(x-2)^{3/2}}{(x^2-4x+4)^{1/4}} = \frac{(x-2)^{3/2}}{((x-2)^2)^{1/4}}$$

Mivel a Maple nem teszi föl, hogy $x \geq 2$, nem tudja egyszerűsíteni a négyzetgyökös kifejezést. A négyzetgyökökre vonatkozó általános egyszerűsítés végrehajtása a `symbolic` kulcsszó megadásával kényszeríthető ki.

```
> radical_formula = simplify( radical_formula, symbolic );
```

$$\frac{(x-2)^{3/2}}{(x^2-4x+4)^{1/4}} = x-2$$

A lényeg az, hogy egyetlen eljárás, a **simplify** különböző típusú egyszerűsítéseket hajt végre: trigonometrikus egyszerűsítéseket, logaritmikus és exponenciális függvények egyszerűsítéseit és racionális kitevős hatványok egyszerűsítését. Másrészt, a mintaillesztés és a transzformációs szabályok koncepciója jelenleg nem eléggé épült be a rendszerbe. A Maple-ben nehézkes a globálisan végrehajtandó matematikai transzformációk programozása.

Számos helyen maga a Maple dönt a követendő útról. Négy példát említünk. Mátrix determinánsának kiszámítása kis mátrixok esetén kifejtéssel történik, különben a Gauss-eliminációt használja a rendszer. A Maple-nek lényegében három numerikus eljárása van véges intervallumon végzett határozott integrál kiszámítására; az alapértelmezés szerinti a Clenshaw–Curtis kvadratúra, ám ha (a közeli szingularitások miatt) lassú a konvergencia, akkor a rendszer megpróbálja kiküszöbölni a szingularitást, vagy átvált egy adaptív duplán-exponenciális kvadratúra-eljárásra. Ha kissebb pontosság is elegendő (pl. `Digits` ≤ 15), használhatunk adaptív Newton–Cotes módszert is. Nota bene, az általánosított sorfejtés és a változó transzformáció két olyan technika, amit a Maple az **evalf/int** eljárásban is alkalmaz analitikus integrandus esetén a szingularitások kezelésére. Az érdeklődő Olvasónak [71, 75] áttekintését javasoljuk. Jelenleg hatféle algoritmust kódoltak az **fsolve** eljárásban: a Newton, a szelő, a dichotomikus, az inverz parabolikus interpolációt végző, (egyenletrendszerekre) a Jacobi mátrix közelítésével számoló és (ismét csak rendszerekre) egy parciális behelyettesítési eljárást. A Maple fölhasználójaként nem kell törődnünk ezekkel a részletekkel; maga a rendszer megtalálja a helyes utat. Ez a szemléletmód teszi a Maple-t a számítógépen végzett matematikai számítások könnyen megtanulható és könnyen kezelhető eszközévé.

2.

Az első lépések: számolás számokkal

Ebben a fejezetben azokat az alapvető ismereteket tárgyaljuk, amelyek a Maple használatához, az on-line help eléréséhez és a számokon végzett műveletek végrehajtásához szükségesek. A rendszerrel való komunikációra hosszasabban a 4. fejezetben térünk ki. A Maple által támogatott számtestek áttekintése egyben „fájdalommentes" bevezetés a rendszer belső adatábrázolásába is.

Föltételezzük, hogy az Olvasó a Maple V Release 4-nek az X Window System alatti munkalapos fölhasználói felületű változatát használja. A továbbiakban ismertetett számításokat 75 Mhz-es Super SPARC II processzoros, 64 Mbyte memóriával és 128 Mbyte swap területtel ellátott, SunOS 4.1.3 operációs rendszerű SUN SPARCstation 20 M71 típusú munkaállomáson végeztük. A könyvben leírtak többsége ugyanúgy igaz más gépek és/vagy más operációs rendszerek esetén is, de előfordulhatnak az implementációtól függő különbségek, például a Maple elindítása, megszakítása vagy leállítása, fájlok írása és olvasása, valamint az eredmények kirajzolása területén. A rendszerfüggő jellegzetességekről olvassuk el a szoftverrel adott dokumentációt. Azt is föltételezzük, hogy az Olvasó munkaállomást vagy X terminált használ, és tisztában van azzal, hogyan kell bejelentkezni számítógépére.

2.1. A kezdetek

A Maple munkalapos felületű változatát a Unix shellből az **xmaple** paranccsal indíthatjuk el. Ha ez nem működne, kérjünk segítséget a helyi szakértőktől, és nézzük meg a dokumentációt. Ha sikeresen betöltődött a munkalapos felületű Maple, a képernyőn egy üres munkalap, és ennek elején a prompt karakter (általában „>") jelenik meg, jelezve, hogy a rendszer parancsra vár. Ekkorra már automatikusan végrehajtódtak a megfelelő inicializáló fájlokban (home könyvtárunk `.mapleinit` fájljában és/vagy a Maple szoftvert tartalmazó könyvtár `src` alkönyvtárának `init` fájljában) található parancsok.[1] Ezek után a közönséges zsebszámológépekhez hasonlóan használhatjuk a Maple-t:

```
> 2 + 100 / 5^2 * 3 ;
```

$$14$$

```
> 5! / 21 ;
```

$$\frac{40}{7}$$

Az ismerős aritmetikai műveletek mind rendelkezésünkre állnak: az összeadás (+), a szorzás (*), az osztás (/), a hatványozás (^ vagy **), a faktoriális művelet (**factorial** vagy !) és a !! dupla faktoriális. A szokásos precedencia-szabályok érvényesek, de kétséges esetekben vagy a könnyebb érthetőség kedvéért kitehetjük a zárójeleket. Ha a Maple-t zsebszámológépként használjuk, ügyeljünk az alábbiakra:

- Mindegyik parancsot pontosvesszővel vagy kettősponttal kell lezárni. Sose felejtsük el ezeket, különben a rendszer további input jelekre várakozik. A parancs pontosvesszővel való lezárása arról informálja a Maple-t, hogy a parancs beírása befejeződött, végre kell hajtani, az eredményt a képernyőn meg kell jeleníteni. A kettőspontot akkor alkalmazhatjuk, ha szükségünk van egy részeredményre, de nem akarjuk látni az ennek megfelelő outputot. Az utasítás végrehajtásának folyamatát *kiértékelésnek* nevezzük.

- A Maple csak akkor kezd foglalkozni egy input sorral, ha egy *új sor* (new line) karaktert kap. Ez lehet a *kocsi vissza* (carriage return) vagy a *soremelés* (line feed) karakter, amelyet a RETURN, illetve az ENTER billentyű lenyomásával adhatunk meg. A parancs kiértékelése után a Maple új prompt jelet ír ki a következő sor elejére, ezzel jelezve, hogy újra inputra várakozik.

[1] Az inicializáló fájlok helye és neve erősen rendszerfüggő. (*A Fordító megjegyzése.*)

A Maple input módját szemléltetik az alábbi példák, különös tekintettel a pontosvessző és a kettőspont használatára:

```
> 2*5; 2^5: 100/4; 100
> /6
> ;
```

$$10$$

$$25$$

$$\frac{50}{3}$$

```
> 12345\
> 6789;
```

$$123456789$$

Jól látható, hogy egy sorban egynél több utasítás is megadható. A Maple ezeket egyesével kezeli, mintha külön-külön adtuk volna meg őket. A jobb olvashatóság kedvéért egy utasítást több sorba is tördelhetünk, illetve szóközöket is használhatunk. Arra azonban ügyeljünk, hogy ez nem mindenhol megengedett. Például az 1000000 számot így nem gépelhetjük le: 1 000 000, de hármas csoportokba oszthatjuk a jegyeket a következő módon: 1\000\000. A „\" fordított törtvonalat (backslash) a rendszer *folytató karakterként* használja, melyet elhagy a feldolgozás során. Ha egy utasítást kettő vagy több sorban adunk meg, az újabb sorok elkezdésekor a Maple figyelmeztet bennünket, hogy hiányos utasítást kapott, vagy elfeledkeztünk a pontosvesszőről. A parancsokat az új sor karaktereknél részekre bontja, kivéve, ha a sort „\"-sel zárjuk le. Ebben az esetben mind a fordított törtvonalat, mind pedig az új sor karaktert a rendszer ignorálja.

Ha a Maple szintaktikus hibák miatt nem tudja értelmezni az input sort, erről is tájékoztat:

```
> this is a line Maple does not understand

Syntax error, missing operator or `;`
```

A munkalapos felület használatakor villogó kurzort látunk annál a jelnél, amelynek beolvasásakor a rendszer a hibát észlelte. A Maple nem mond ennél többet. A hiba okát nekünk kell megtalálni, addig a Maple az input korrekciójára várakozik. A munkalapos felületen az input kijavítható vagy kiegészíthető, ha a kurzort a megfelelő sorra visszük, majd elvégezzük az összes szükséges változtatást és/vagy pótlást. A karakteres felületű Maple, amelyet a Unix shellből a **maple** paranccsal indíthatunk el, beépített sorszerkesztőt tartalmaz, amely mind vi-, mind emacs-szerű módon használható, és az utoljára begépelt 100 sor szerkesztését teszi lehetővé.

A Maple-ből való kilépéshez kattintsunk rá a File és az Exit menüpontokra. A parancs végrehajtása előtt a rendszer még megerősítést kér. Alternatív megoldásként használhatjuk a megfelelő gyorsítóbillentyűket. Közvetlenül is kiléphetünk a Maple-ből a **quit**, **stop** vagy a **done** utasítás és az új sor karakter beírásával.

Az éppen zajló hosszabb számításokat a Maple ablak eszközsorának Stop gombjára kattintva szakíthatjuk félbe. Ekkor az operációs rendszer egy *interrupt* karaktert (általában CONTROL-C) kap. Nem mindig tudjuk azonnal félbeszakítani a számolást, de miután a rendszer fölismerte a megszakítást, beírhatunk újabb parancsot.

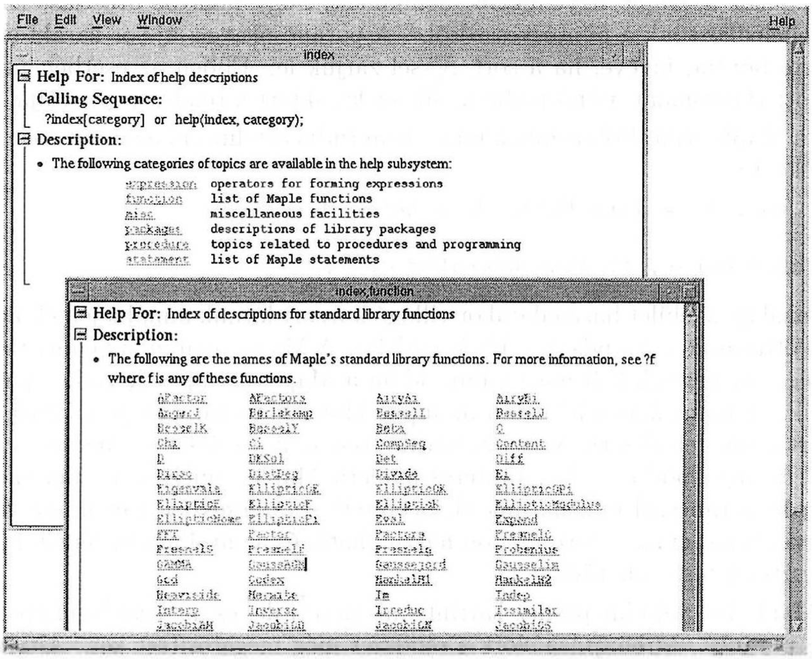
A Maple rendszert újraindíthatjuk belülről is a **restart** paranccsal.

```
> restart;
```

Ez az utasítás törli a Maple belső memóriáját, újraolvassa az inicializáló fájlokat, szóval úgy működik, mintha a rendszert valóban teljesen „tiszta lappal", elölről indítanánk. Csak az interfész változók beállításai maradnak meg.

2.2. A Help rendszer

Ha a Maple munkalap Help gombjára kattintunk, a rendelkezésünkre álló help szolgáltatásokat mutató menü jelenik meg. Kiindulásul érdemes a help témaköreit leíró oldalt választani, amely a Contents menüponttal vagy egy C betű bevitelével érhető el. A 2.1. ábra a képernyőn megjelenő oldalt mutatja. Ha a function-ra kattintunk, a hiperlinket követve a standard könyvtári függvények leírásának tárgymutatójához jutunk.

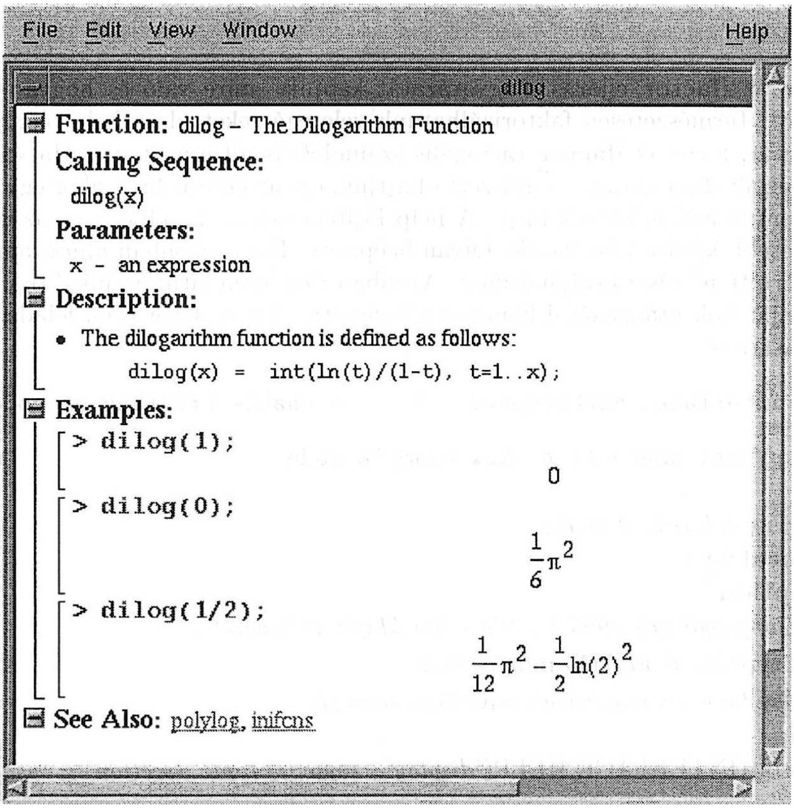


2.1. ábra: A help témakörök és a standard könyvtári függvények tárgymutatója

További információkat a kiválasztott függvény nevére, a dilogaritmus esetében például a `dilog`-ra kattintva kaphatunk. A függvény használatára vonatkozó tényleges információt tartalmazó help ablak a 2.2. ábrán látható. Ugyanez az ablak jelenik meg akkor is, ha a

```
> ?dilog
```

parancsot gépeljük be. A standard függvények tárgymutatója azért hasznos, mert lehetővé teszi, hogy a [160] „Maple Handbook" lapozgatása nélkül eligazodjunk, és megtaláljunk dolgokat. A *?<topic>* szintaxisnak megfelelően gyorsabban megkaphatjuk a szükséges információkat, ha tudjuk, melyik parancsot keressük, és csak ismereteinket szeretnénk fölfrissíteni, vagy példákat akarunk látni.



2.2. ábra: A `dilog` Maple függvény help oldala

A rendelkezésünkre álló on-line help konkrét példájaként kérjünk segítséget az egészek faktorizálásáról:

```
> ?ifactor
```

A megfelelő help üzenet jól illusztrálja a szintaxist, az adattípusokat és a függ-

vényeket leíró Maple oldalak általános alakját. Az alábbiakban a help ablak tartalmát több részre vágtuk, és azt is megmutatjuk, hogyan készíthetünk ilyen kivonatokat egy Maple szekció során:

```
> info( ifactor ); usage( ifactor );

Function: ifactor - integer factorization

Calling Sequence:
   ifactor(n)
   ifactor(n, method)

Parameters:
   n      - integer or a rational
   method - (optional) name of base method for factoring
```

Először az **ifactor** eljárás magyarázatát kapjuk: mire való és hogyan használható. Természetesen faktorizálhatunk vele egészeket, de a help oldal arról tájékoztat, hogy az **ifactor** racionális számokra is alkalmazható. Ha valamelyik speciális faktorizációs módszert óhajtjuk, ezt az eljárás hívásakor egy extra argumentummal adhatjuk meg. A help fájlban ezután az eljárás és az opciók részletesebb leírása következik. Olyan beépített eljárás azonban nincs, amellyel a „Description" részt kivághatnánk. Azonban rövidesen látni fogjuk, hogy a hasonló eljárások utánzásával könnyen kiterjeszthető a szoftver ezen feladat végrehajtására is:

```
> interface( verboseproc = 3 ):  # enable printing of code

> print( usage ); # show example code
```

proc(x::{*indexed*, *string*})
  local *topic*;
  option
  '*Copyright (c) 1995 by Waterloo Maple Software*'
  if type(x, *string*) **then** *topic* := x
  else *topic* := traperror(convert(x, *string*))
  fi;
  print(INTERFACE_HELP('*display*', '*topic*' = *topic*, '*section*' = '*usage*'))
end

Ebben a szellemben definiálunk egy **synopsis** eljárást, és ki is próbáljuk az **ifactor** eljáráson[2]:

[2] Az angol eredetiben szereplő „Synopsis" helyett a Maple Linuxos verziójában a help oldal megfelelő részén található alcím „Description", de az eljárás ugyanígy használható. (*A Fordító megjegyzése.*)

```
> synopsis := proc( x::{indexed,string} )
>    local topic;
>    if type(x,string) then topic := x
>    else topic := traperror( convert(x,string))
>    fi;
>    print( INTERFACE_HELP(
>      'display', 'topic' = topic, 'section' = 'synopsis' ) )
>  end:
> synopsis( ifactor );

Description:

- ifactor returns the complete integer factorization of n.

- The answer is in the form:
  u * ``(f1)^e1 * ... * ``(fn)^en such that
  n = u * f1^e1 * ... * fn^en where u equals sign(n),
  f1, ..., fn are the distinct prime factors of n,
  and e1, ..., en are their multiplicities (negative in
  the case of the denominator of a rational).

- The expand function may be applied to cause the factors
  to be multiplied together again.

- If a second parameter is specified, the named method
  will be used when the front-end code fails to achieve
  the factorization. By default, the Morrison-Brillhart
  algorithm is used as the base method. Currently
  accepted names are:

    'squfof'  - D. Shanks' undocumented square-free
                factorization;
    'pollard' - J.M. Pollard's rho method;
    'lenstra' - Lenstra's elliptic curve method; and
    'easy'    - which does no further work.


- If the 'easy' option is chosen, the result of the
  ifactor call will be a product of the factors that
  were easy to compute, and a name _c.m indicating
  an m-digit composite number that was not factored.

- The pollard base method accepts an additional optional
  integer: ifactor(n,pollard,k), which increases the
  efficiency of the method when one of the factors is
  of the form k*m+1.
```

A help oldalon ezután néhány példa következik. Sokszor már ezekből kiderül, hogyan oldható meg valamely feladat a Maple-lel.

```
> example( ifactor );

Examples:
> ifactor( 61 );
                                    (61)


> ifactor( 60 );
                                      2
                                   (2)  (3) (5)

> ifactor( -144 );
                                      4   2
                                  -(2)  (3)

> expand(");
                                    -144

> ifactor( 60, easy );
                                      2
                                   (2)  (3) (5)

> ifactor( 4/11 );
                                         2
                                      (2)
                                      ----
                                      (11)


> n := 8012940887713968000041:
> ifactor( n, easy );
                             (13) (457) _c19

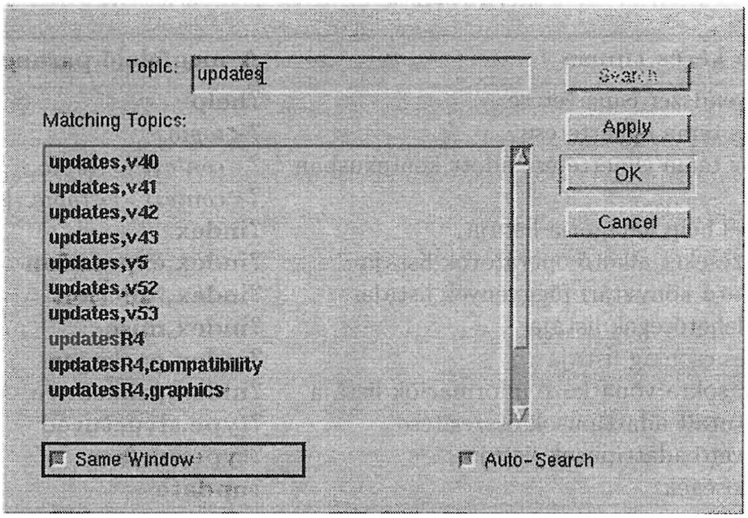
> ifactor( n );
                  (13) (457) (2847639359) (473638939)
```

Végül a kapcsolódó fogalmakra utal a Maple:

```
> related( ifactor );

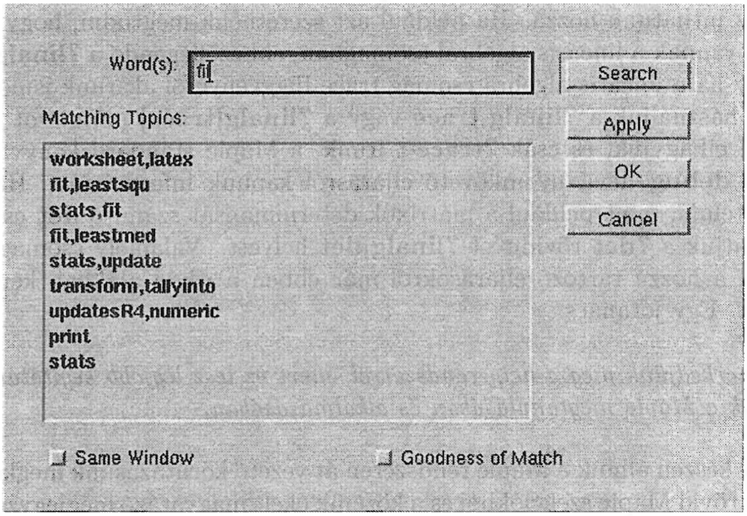
See Also: ifactors, isprime, factor, type[facint]
```

Az **info**(*<topic>*) és az **example**(*<topic>*) helyett használhatjuk a `??`*<topic>*, illetve a `???`*<topic>* rövid alakot. A help rendszerben témakörök szerint is kereshetünk. Ha a Help menüből a Topic Search-öt választjuk, a 2.3. ábrán látható dialógusbox jelenik meg. Az `updates` szót beírva azon témakörök listáját kapjuk, melyek neve ezzel a szóval kezdődik.



2.3. ábra: Témakörök szerinti keresés

Adott szavakat tartalmazó help témakörök között más módon is kereshetünk. A 2.4. ábra a |Help| menü |Full Text Search| opciójának megfelelő dialógusboxot mutatja. A `fit` sztringet tartalmazó témaköröket írattuk ki.



2.4. ábra: Szöveg szerinti keresés

Láttuk tehát, hogy rengeteg segítséget kaphatunk a standard könyvtári függvényekről magától a rendszertől is. A 2.1. táblázatban összegeztük a Maple help rendszerének lehetséges eléréseit.

A help kérés típusa	A megfelelő parancs
a help rendszer ismertetése	**?help**
speciális téma ismertetése	?*<topic>*
speciális téma ismertetése adott kontexusban	?*<context>* *<topic>*
	?*<context>*[*<topic>*]
az összes help kategória listája	**?index**
a kifejezéseket alkotó operátorok listája	**?index,expression**
a standard könyvtári függvények listája	**?index,function**
vegyes lehetőségek listája	**?index,misc**
az összes csomag listája	**?index,packages**
az eljárásokra vonatkozó információk listája	**?index,procedure**
a strukturált adattípusok összesítése	**?type,structured**
az alapvető adattípusok listája	**?type,surface**
új lehetőségek	**?update**

2.1. táblázat: A Maple on-line help rendszere

A **?index** paranccsal kaphatjuk meg a help kategóriák listáját. Ez többek között a *function*, a *misc* és a *packages* kategóriákat tartalmazza. A Maple összes standard könyvtári függvényét fölsorolja a **?index,function** utasítás. A rendszer ritkábban használt lehetőségeit ismerhetjük meg a **?index,misc** paranccsal. A **?packages** utasítás beírásával a Maple-ben rendelkezésünkre álló csomagok listájához juthatunk hozzá. Ha például azt szeretnénk megtudni, hogy milyen eljárások vannak a lineáris algebrai csomagban, ehhez elegendő a **?linalg** utasítás. Vagy ha a lineáris algebra csomag trace függvényéről akarunk ismereteket szerezni, használjuk a **?linalg,trace** vagy a **?linalg[trace]** parancsot. Ha az első részt elhagyjuk, és csak **?trace**-t írunk, a Maple standard könyvtárában található **debug** nevű nyomkövető eljárásról kapunk információt. Ha a kérés egyértelmű, mint például a mátrixok determinánsát számító **det** esetében, használhatjuk a **?det** rövidítést **?linalg,det** helyett. Valamely csomag betöltése után a hozzá tartozó eljárásokról már ebben a rövid alakban kérhetünk segítséget. Egy jótanács:

> *Ismerkedjünk meg a help rendszerrel, mert ez lesz legjobb segítőtársunk a Maple megtanulásában és alkalmazásában.*

Most már készen állunk a Maple rendszeren át vezető körutazásunk megkezdésére, amely rövid Maple szekciókból és a közéjük ékelt magyarázó megjegyzésekből fog állni. Ha mást nem mondunk, azt tételezzük föl, hogy példáinkban a Maple már elfelejtette az előző példák teljes történetét, mintha új Maple szekciót indítottunk volna. Alaposan nézzük át a példákat, próbáljuk ki őket a Maple segítségével, és kedvünk szerint változtassunk is rajtuk. Csak akkor szerezhetünk megfelelő ismereteket és jártasságot a számítógépes algebra használhatóságáról és korlátairól, ha közvetlenül szembesülünk magával a számítógépes algebrai

rendszerrel. Ugyanezen megfontolásból minden egyes fejezetet gyakorlatokkal zárunk. Dolgozzuk ki ezeket részletesen, hogy egyre gyakorlottabb Maple fölhasználókká válhassunk.

2.3. Egész és racionális számok

Igazi számítógépes algebrai rendszer lévén a Maple is idegenkedik közelítések alkalmazásától az aritmetikai számítások során. A zsebkalkulátorokkal ellentétben a Maple spontán módon soha nem alakít át aritmetikai kifejezéseket decimális számmá.

Két egész szám nemnulla nevezőjű hányadosát is csak egyszerűsíti. Pontosabban szólva, a racionális számokat automatikusan egyszerűsíti a számláló és a nevező legnagyobb közös osztójával. Ahhoz, hogy a racionális számokon pontos aritmetikával tudjon számolni, a Maple-nek nagyon nagy egészeket is tudnia kell kezelni.

```
>  5 / 81491528324789773434561
>  - 101 / 10198346916138403856439;
```

$$\frac{-908850291802563899734274}{92342097398058352671328089792352035178332031}$$

```
>  number := 4^(4^4);
```

$$number := 134078079299425970995740249982058461274\backslash$$
$$79365820592393377723561443721764030073546976801\backslash$$
$$87429816690342769003185818648605085375388281194\backslash$$
$$6569946433649006084096$$

```
>  length( number ); # number of digits
```

$$155$$

Figyeljük meg, hogy a Maple magától kirakja a backslash karaktert, ezzel jelezve, hogy az output a következő sorban folytatódik.

Természetesen létezik olyan legnagyobb egész szám, amely még reprezentálható a Maple-ben, de ennek értéke sokkal nagyobb, mint a legtöbb programozási nyelvben. A jegyek számának felső korlátja 32 bites gépeken $2^{19} - 9 = 524279$. Ez a korlátozás megkerülhető, lásd a [83]-at. Annak titka, hogyan képes a Maple 10^{524279} nagyságú egész számokkal számolni, az egészek belső ábrázolásában rejlik. A legtöbb programozási nyelv a hardver adta lehetőségeket használja az úgynevezett *egyszeres pontosságú* egészekkel való számolásnál. Ez a megközelítés az egész értékeket az egy szóban ábrázolható egészek tartományára korlátozza. A Maple rendszerben *többszörös pontosságú egész aritmetikát* implementáltak oly módon, hogy egy-egy egész szám belső ábrázolására a memóriában több egymás utáni szót használtak föl, lásd [39, 75]. Ezt a lineáris listát *dinamikus adatvektornak* nevezzük, melynek *hosszát* a felhasznált szavak számával definiáljuk. A Maple egész számokat reprezentáló belső adatstruktúrája a következő 2.5. ábrán látható.

intpos	integer i_0	integer i_1	......	integer i_n

2.5. ábra: A pozitív egészek belső ábrázolása

Ennek a vektornak első szava kódolja az adatstruktúra leírásához szükséges összes információt: jelzi, hogy a vektor pozitív egészet reprezentál, valamint, hogy a vektor hossza $n+2$. A következő $n+1$ szó az $i_0, i_1, i_2, \ldots, i_n$ egyszeres pontosságú nem negatív egész számokat tartalmazza. Ha B jelöli a számítógép belső számábrázolásának alapszámát, akkor a fönti vektor az

$$i_0 + i_1 B + i_2 B^2 + i_3 B^3 + \cdots + i_n B^n$$

egészet reprezentálja. A Maple B-ként 10-nek azt a legnagyobb hatványát használja, amelyre B^2 még egyszeres pontosságú aritmetikában ábrázolható (32 bites gép esetén $B = 10^4$). Mivel a vektor hossza nem előre rögzített fix érték, hanem dinamikusan választható, ezért a Maple rendszerben nagyon nagy egészek reprezentálhatók. Az egyetlen megszorítást az jelenti, hogy a számot ábrázoló vektor szavainak számát a dinamikus adatvektor első szavában kell ábrázolnunk. A Maple 17 bitet használ a vektor hosszának megadására, ebből adódik a $2^{19} - 9 = 4\left((2^{17}-1)-1\right) - 1$ „mágikus szám". A Maple elég intelligens ahhoz, hogy egy számról előre eldöntse, vajon ábrázolható-e:

```
> 123456789 ^ 987654321;

Error, object too large
```

A Maple-ben több olyan eljárás is van, amelyek segítségével egészeken számításokat végezhetünk. Nézzünk néhány példát:

```
> number := 10^29 - 10^14 - 1;
```

$$number := 99999999999999899999999999999$$

```
> isprime( number ); # check whether the number is prime
```

false

A Maple prímtesztelő algoritmusának leírását [200]-ban találjuk.

```
> settime := time(): # start timing
> ifactor( number ); # factorize the integer
```

$$(61)\,(223)\,(13166701)\,(97660768252549)\,(5717)$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 2.040\,seconds$$

```
> nextprime( number ); # determine the next largest prime
```

$$99999999999999900000000000157$$

```
> # integer approximation to the square root
> isqrt( number );
```

$$316227766016838$$

Az egészek faktorizálása rendkívül időigényes feladat, ezért bizonyára ismerni szeretnénk a számításokhoz fölhasznált időmennyiséget. Ahogy az előbb láttuk, ez a Maple **time** eljárásával kapható meg, amely a Maple szekció kezdete óta a számításokra összesen elhasznált időt írja ki (másodpercekben). Írjuk be közvetlenül a számítás elkezdése előtt, illetve befejezése után a **time**( ) parancsot. A két számérték különbsége adja meg az adott számítás végrehajtási idejét.

Másik alternatíva a **time**(<*expression*>) parancs használata, ezzel az argumentumában szereplő kifejezés kiértékeléséhez szükséges időt kapjuk. Például

```
> time( assign( factored_number, ifactor(3!!!) ) );
```

$$.999$$

a 3!!! = 6!! = 720! prímfaktorizációját számolja ki, az eredményt a későbbi hivatkozások céljából a `factored_number` változóban tárolja. Itt nem használható a := értékadó operátor, mivel a **time** argumentuma nem lehet (értékadó) utasítás, csak kifejezés.

Az összes előző eljárásban kulcsszerepet játszik az egészek (maradékos) osztása és a legnagyobb közös osztó meghatározása:

```
> a := 1234: b := 56:
> q := iquo(a,b); # quotient of integer division
```

$$q := 22$$

```
> r := irem(a,b); # remainder of integer division
```

$$r := 2$$

```
> testeq( a = q*b + r ); # check identity
```

$$true$$

```
> igcd(a,b); # greatest common divisor of integers
```

$$2$$

Az **igcdex** eljárás egészek kiterjesztett legnagyobb közös osztóját (**ex**tended **g**reatest **c**ommon **d**ivisor of **i**ntegers) számolja: az a és b egészekhez olyan s és t egészeket határoz meg, amelyekkel $a\,s + b\,t = \gcd(a, b)$:

```
> igcdex( a, b, 's', 't' );
```

$$2$$

```
> 's' = s, 't' = t, 'a*s + b*t' = a*s + b*t;
```

$$s = 1,\ t = -22,\ a\,s + b\,t = 2$$

Az előző példákban az s, t és az $a\,s + b\,t$ körüli idézőjelek a változók kiértékelésének elnyomására szolgálnak. Ennek részletesebb magyarázata a következő fejezetben található meg.

A moduláris aritmetika fontos szerepet játszik az egészek faktorizálásában és a prímtesztekben, lásd [117]. A **mod** operátor alkalmazásának eredménye a (nemnulla) n egész szerinti moduláris aritmetikában a $0, 1, 2, \ldots, |n| - 1$ sorozat valamelyik tagja. Ha ehelyett inkább a

$$-\left\lfloor \frac{|n|-1}{2} \right\rfloor, \ldots, -1, 0, 1, \ldots, \left\lfloor \frac{|n|}{2} - 1 \right\rfloor, \left\lfloor \frac{|n|}{2} \right\rfloor$$

közé eső, a 0 körül szimmetrikusan elhelyezkedő értékeket preferálunk, ezt előre meg kell mondanunk.

```
> 1/2345 mod 6;
```

$$5$$

```
> `mod` := mods: 1/2345 mod 6;
```

$$-1$$

A Maple moduláris aritmetikájának egyik hátránya, hogy egy lépésben nem írható elő, hogy az összes további számítást valamely fix modulus szerint végezze a rendszer. Minden parancsnál hívnunk kell a **mod** rutint.

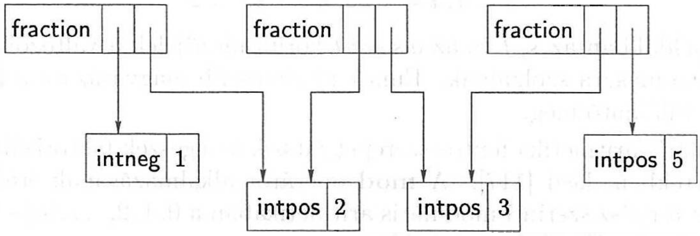
Mint korábban megjegyeztük, a Maple a racionális számokat magától egyszerűsíti egyértelműen meghatározott standard alakjukra. A rendszer automatikusan egyszerűsít a számláló és a nevező legnagyobb közös osztójával, és azt is biztosítja, hogy a nevező pozitív legyen. A racionális számok legkompaktabb standard formája a (*nominator*, *denominator*) alakú, pozitív nevezőjű számpárokkal való reprezentálás, amelyet *kanonikus alaknak* nevezünk. Nem valószínű, hogy az Olvasó az $\frac{1}{3}$-ot inkább a

$$\frac{-41152263041152263041152263041152263041152263}{-123456789123456789123456789123456789123456789}$$

alakban szeretné ábrázolni. Azt gondolhatnánk, hogy a Maple a racionális számok belső ábrázolására olyan három komponensű adatvektorokat használ, amelyekben az első komponens a vektor fajtáját, a második a számlálót, a harmadik pedig a pozitív nevezőt reprezentálja. A Maple tervezői azonban más megoldást választottak, mivel a rendszer memória-használatát a lehető leghatékonyabbá akarták tenni. A következő szabályból indultak ki (lásd [36]):

A Maple minden (rész)kifejezést csak egyszer tárol.

Ezért a racionális számot reprezentáló adatvektorban a két utolsó komponens nem a számlálót és a nevezőt reprezentáló többszörös pontosságú egészeket, hanem csak rájuk mutató pointereket tartalmaz. Így ha egy egész szám esetleg több törtben elő is fordul, csupán egyetlen helyen kell tárolni a memóriában. Ha például egy Maple szekcióban a $-\frac{1}{2}$, $\frac{2}{3}$ és a $\frac{3}{5}$ racionális számokat használjuk, a belső reprezentáció a 2.6. ábrán látható alakú lesz:



2.6. ábra: A $-\frac{1}{2}$, $\frac{2}{3}$ és a $\frac{3}{5}$ törtek belső ábrázolása

2.4. Irracionális és lebegőpontos számok

Az előző részben azt láttuk, hogy a Maple a racionális számokat automatikusan egyszerűsíti. Általában azonban a rendszer csak akkor hajt végre egy számítást, ha erre kimondottan utasítjuk:

```
> 25^(1/6);
```

$$25^{1/6}$$

```
> simplify(");
```

$$5^{1/3}$$

```
> evalf("");
```

$$1.709975947$$

```
> convert( """, float );
```

$$1.709975947$$

Ebben a példában a **simplify** és az **evalf** eljárásokkal dolgoztattuk meg a Maple-t. A **simplify**(") parancsban az idézőjel az előzőleg kiértékelt kifejezésre, esetünkben a $25^{1/6}$-ra való hivatkozást jelent. Két idézőjel az utolsó előtti, három az azt megelőzően kiértékelt kifejezést adja; háromnál több idézőjel nem használható. A fönti értelemben alkalmazott idézőjeleket *ditto operátornak* is nevezzük.

Arra számíthattunk volna a fenti példában, hogy a Maple rögtön a köbgyök közelítő értékét adja meg, ez azonban ellentmondana a pontos aritmetika elvének. Lehet, hogy a fönti köbgyök harmadik hatványát is ki akarjuk számolni és a pontos eredményt szeretnénk megkapni. Közelítő lebegőpontos aritmetikával eltérő eredményre jutnánk.

```
> 25.0^(1/6);
```

$$1.709975947$$

```
> "^6;
```

$$25.00000003$$

A Maple minden tizedespontot tartalmazó számot lebegőpontos számnak tekint, és a továbbiakban így is számol vele. Gondoskodik a szükséges automatikus típuskonverziókról (például egészről lebegőpontosra) is:

```
> 90005*0.15;
```

$$13500.75$$

A Maple-ben a lebegőpontos számok „megfertőzik" a további számításokat, ezért `exp(1.)` már automatikusan `2.71828....`-ra értékelődik ki.

A lebegőpontos számokra többféle jelölést alkalmazhatunk. A `0.000001` például megadható `0.1*10-5`, `1E-6` és `Float(10,-6)` formában is. A lebegőpontos számok Maple-beli belső ábrázolására emlékeztet a `Float`(*mantissa*, *exponent*) alakú utóbbi jelölés: ez olyan adatvektor, mely a `Float` fejlécből, továbbá a többszörös pontosságú egész számokkal megadott mantisszára és kitevőre mutató két pointerből áll. Ez a vektor a $mantissa \times 10^{exponent}$ számot ábrázolja. Más szavakkal, a lebegőpontos szám szignifikáns jegyeit a mantisszában tároljuk, míg az általános nagyságrendjét a kitevő adja meg. A lebegőpontos számok belső ábrázolásának következményeként ezen számok pontossága megegyezik a Maple-ben ábrázolható legnagyobb egész szám számjegyeinek számával. Az egész kitevőkön végzett műveleteket (összeadás, szorzás, ...) azonban a C nyelven írták meg. Ez azt jelenti, hogy a kitevő-aritmetika a C egész aritmetikájára van megszorítva.

A lebegőpontos aritmetika pontosságát a `Digits` nevű Maple változó beállításával határozhatjuk meg, ennek alapértelmezése 10. Több olyan függvény van, amelynek eredményét a Maple lebegőpontos aritmetikával számolja ki, ezek közül a legfontosabb az **evalf** (**eval**ute using **f**loating-point arithmetic):

```
> evalf( sqrt(2) );
```

$$1.414213562$$

```
> Digits;
```

$$10$$

```
> Digits := 20:  evalf( sqrt(2) );
```

$$1.4142135623730950488$$

```
> evalf( Pi, 150 );
```

3.14159265358979323846264338327950288419716939937\
751058209749445923078164062862089986280348253421\
170679821480865132823066470938446095505822317253\
535940813

Az **evalf** eljárás első argumentumát közelíti a második argumentumaként megadott számú jeggyel. Ha csak egy argumentummal hívjuk, akkor a Maple a lebegőpontos aritmetikai számításokat a `Digits` értékének megfelelő számú jeggyel hajtja végre. [188]-ban részletes leírást talál az Olvasó a Maple lebegőpontos számítási modelljéről és ennek következményeiről.

A Maple természetesen ismer néhány matematikai konstanst, például a γ Euler–Mascheroni konstanst és a π számot. A rendszer által ismert konstansokat a 2.2. táblázatban soroltuk föl.

A matematikai konstans	Maple neve	(Közelítő) értéke
π, az egységkör területe	Pi	3.141592654
A C Catalan-szám $= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$	Catalan	0.9159655942
A γ Euler–Mascheroni konstans $= \lim_{n\to\infty} \left(\left(\sum_{k=1}^{n} \frac{1}{k} \right) - \ln n \right)$	gamma	0.5772156649
a *true*, *false*, *fail* logikai értékek	true,false, FAIL	
∞	infinity	

2.2. táblázat: A Maple matematikai konstansai

A Maple konstansok nevei védettek, nehogy véletlenül fölülírjuk őket:

```
> Pi := 3.14;

Error, attempting to assign to `Pi` which is protected
```

Definiálhatunk új védett nevű szimbolikus konstansokat is:

```
> electron_rest_mass := 9.109558 * 10^(-31) * kg;
```

$$electron_rest_mass := .91095580000000000000\, 10^{-30}\, kg$$

```
> protect( 'electron_rest_mass' ):
> electron_rest_mass := 5.48593 * 10^(-4)
> * atomic_mass_units;

Error, attempting to assign to `electron\_rest\_mass` which is
protected
```

Ha új értéket akarunk a konstanshoz rendelni, először törölni kell védettségét az **unprotect**-tel:

```
> unprotect( 'electron_rest_mass' ):
> electron_rest_mass := 5.48593 * 10^(-4)
> * atomic_mass_units;
```

$$electron_rest_mass := .00054859300000000000000\ atomic_mass_units$$

```
> protect( 'electron_rest_mass' ):
```

Másik alternatívaként a **macro** eljárás alkalmazásával rendelhetjük hozzá az `electron_res_mass` védett névhez a 9.109558×10^{-31}kg értéket. Ekkor az `electron_res_mass` csupán egy speciális érték olyan rövidítése, amelynek nem adhatunk véletlenül új értéket:

```
> macro( electron_rest_mass = 9.109558 * 10^(-31) * kg ):
> electron_rest_mass;
```

$$.9109558000000000000\,10^{-30}\,kg$$

Az utóbbi mechanizmus jól alkalmazható a természetes alapú logaritmus e alapszámára. A Maple ezt `exp(1)`-gyel jelöli. De elég fárasztó `exp(1)`-et írni valahányszor az e-re van szükségünk. Azon túl, hogy a munkalapjainkon inkább e-t szeretnénk látni, számolás közben is ezt a rövid jelölést szeretnénk használni. Nos, ennek eléréséhez a következőképpen járhatunk el:

```
> protect( 'e' ):

> macro( e = exp(1) ):

> ln(e);
```

$$1$$

A zsebszámológépeken használhatjuk az exponenciális, a természetes alapú logaritmus, a trigonometrikus és más matematikai függvényeket. Mindezek megtalálhatók a Maple-ben is, de a rendszer rajtuk kívül még sokkal több függvényt is nyújt. A 2.3. táblázatban található a gyakoribb matematikai függvények fölsorolása a Maple-ben használatos nevükkel együtt. Részletesebb listát kaphatunk a **?inifcns** (help about **ini**tially known **f**un**c**tio**ns**) paranccsal.

Idézzünk föl néhány definíciót:

a Gamma függvény $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt, \quad \Re(z) > 0$

a Riemann-féle ζ függvény $\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$

A dilogaritmus függvény $\operatorname{dilog}(x) = \int_1^x \frac{\ln t}{1-t}\, dt$

a hibafüggvény $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$

A Maple ismeri ezek közül soknak pontos értékeit. A $\frac{\pi}{8}$, a $\frac{\pi}{10}$ és a $\frac{\pi}{12}$ többszöröseire alkalmazott trigonometrikus függvények például pontos numerikus eredményeket adnak. A Riemann-féle ζ függvény eredménye is pontos, ha 50-nél kisebb páros természetes számra alkalmazzuk. Nagyobb számokra az **expand**-dal explicit módon ki kell fejtetni az eredményt. Ismert, hogy $\lim_{x\to\infty} \operatorname{erf}(x) = 1$ és így tovább.

A matematikai függvény	Maple neve
exponenciális függvény	exp
természetes alapú logaritmus	ln, log
tízes alapú logaritmus	log10
negyzetgyök	sqrt
abszulút érték	abs
trigonometrikus függvények	sin, cos, tan, csc, sec, cot
inverz trigonometrikus függvények	arcsin, arccos, arctan, arccsc, arcsec, arccot
hiperbolikus függvények	sinh, cosh, tanh, csch, sech, coth
inverz hiperbolikus függvények	arcsinh, arccosh, arctanh, arccsch, arcsech, arccoth
hipergeometrikus függvény	hypergeom
Bessel függvények	BesselI, BesselJ, BesselK, BesselY
Gamma függvény	GAMMA
binomiális együtthatók	binomial
poligamma függvény	Psi
Riemann-féle ζ függvény	Zeta
dilogaritmus	dilog
hiba függvény	erf

2.3. táblázat: A Maple által ismert, gyakran használt matematikai függvények

```
> sin(Pi/10), Zeta(2), limit( erf(x), x=infinity );
```

$$\frac{1}{4}\sqrt{5}-\frac{1}{4},\ \frac{1}{6}\pi^2,\ 1$$

```
> Zeta(50) = expand( Zeta(50) );
```

$$\zeta(50) = 39604576419286371856998202/$$
$$285258771457546764463363635252374414183254365234375\pi^{50}$$

Numerikus közelítéseket az **evalf**-fel nyerhetünk.

```
> Zeta(3); evalf(");
```

$$\zeta(3)$$

$$1.202056903159594285 4$$

Még néhány elgondolkodtató példa az egzakt és a lebegőpontos aritmetika viszonyáról:

```
> sin(4) - 2*sin(2)*cos(2); combine(",'trig'); evalf("");
```

$$\sin(4) - 2\sin(2)\cos(2)$$

$$0$$

$$.1\,10^{-19}$$

```
> (1+sqrt(2))^2-2*(1+sqrt(2))-1; simplify("); evalf("");
```

$$(\sqrt{2}+1)^2 - 3 - 2\sqrt{2}$$

$$0$$

$$0$$

Talán tudni szeretné az Olvasó, hogyan képes az **evalf** Maple eljárás megkülönböztetni mindezeket a matematikai függvényeket. Föltárulnak a Maple titkai, ha a `printlevel` Maple változó értékét az alapföltételezés szerinti 1-nél nagyobbra állítjuk be:

```
> printlevel := 5:  evalf( sin(1)+ln(2) ); printlevel := 1:

{--> enter sin, args = 1
<-- exit sin (now at top level) = sin(1)}
{--> enter evalf/sin, args = 1
```

$$x := 1.$$

```
<-- exit evalf/sin (now at top level) = .84147098480789650665}
{--> enter evalf/ln, args = 2
```

$$x := 2.$$

```
<-- exit evalf/ln (now at top level) = .69314718055994530942}
```

$$1.5346181653678418161$$

A példa azt jelzi, hogy minden olyan *func* matematikai függvényhez, amelyre numerikus értékek is meghatározhatók, létezik a Maple könyvtárban egy **evalf/func** nevű eljárás. Valahányszor az **evalf** eljárást alkalmazzuk egy kifejezésre, a rendszer megvizsgálja, hogy milyen függvények fordulnak elő benne, és automatikusan alkalmazza a megfelelő **evalf/func** eljárást. Így a numerikus közelítések kiszámításához elegendő *egyetlen* eljárás nevére emlékeznünk ahelyett, hogy több különböző eljárást kellene fölidéznünk. Látni fogjuk, hogy más területeken, például az integrálásnál, a differenciálásnál és az egyszerűsítésnél is hasonló technikával dolgozik a rendszer.

A Maple-ben létezik egy **evalhf** (**eval**uate using **h**ardware **f**loating-point arithmetic) nevű eljárás is. Ezzel fölgyorsíthatjuk numerikus számításainkat (például függvények gráfjának kirajzolását) a hardver beépített lebegőpontos aritmetikájának használatával. Az eljárás argumentumait a hardver lebegőpontos számábrázolásának megfelelő formára konvertálja, dupla pontossággal kiszámítja az eredményt (ami nagyjából a `Digits`=15 értéknek felel meg), végül ezt a Maple ábrázolásának megfelelő lebegőpontos számmá alakítja. Az **evalhf** használatát a következő módon definiált g függvény értékeinek meghatározásával illusztráljuk:

```
> f := x -> arctan( (2*x^2-1)/(2*x^2+1) ):
> g := (x,y) -> f(x) * f(y):
> g(x,y);
```

$$\arctan\left(\frac{2x^2-1}{2x^2+1}\right)\arctan\left(\frac{2y^2-1}{2y^2+1}\right)$$

Számítsuk ki a függvény értékeit azon az 50×50-es négyzetrácson, amelyet a $[-3,3] \times [-3,3]$ négyzetben ekvidisztánsan elhelyezkedő pontok alkotnak:

```
> settime := time(): # start timing
> for i to 50 do
>   for j to 50 do
>     evalf( g( -3 + 6*i/50, -3 + 6*j/50 ) )
> od od:
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 4.111\, seconds$$

```
> settime := time():
> for i to 50 do
>   for j to 50 do
>     evalhf( g( -3 + 6*i/50, -3 + 6*j/50 ) )
> od od:
> cpu_time := (time()-settime) * seconds; # computing time
```

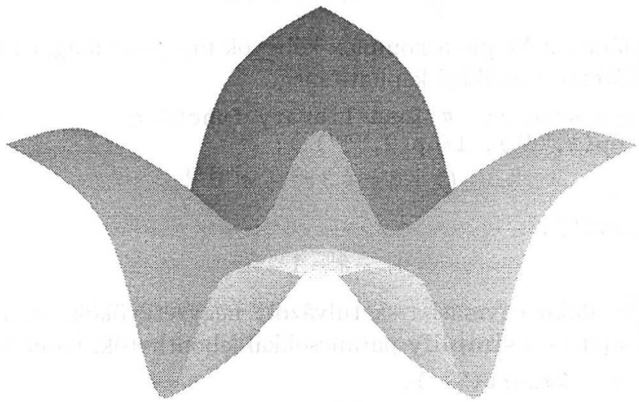
$$cpu_time := .279\, seconds$$

```
> evalf( g(1,1) ), evalhf( g(1,1) );
```

$$.1035234192545466070\!9, .1035234192545466$$

A Maple rajzoló rutinjai, például a **plot3d** szintén a hardver lebegőpontos aritmetikáját használják. Rajzoltassuk ki a g gráfját. (Lásd 2.7. ábra.)

```
> plot3d( g, -3..3, -3..3,
>   grid=[50,50], style=patchnogrid );
```



2.7. ábra: Az $(x, y) \mapsto \arctan - \left(\frac{2\,x^2-1}{2\,x^2+1}\right) \cdot \arctan\left(\frac{2\,y^2-1}{2\,y^2+1}\right)$ függvény grafikonja

2.5. Algebrai számok

Már láttunk példákat radikálokra (gyökökkel fölírt számokra), például egészek négyzetgyökére és köbgyökére. A Maple-nek nem okoz gondot a velük való számolás:

```
> ( 1/2 + 1/2*sqrt(5) )^2;
```

$$\left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^2$$

```
> expand(");
```

$$\frac{3}{2}+\frac{1}{2}\sqrt{5}$$

```
> 1/";
```

$$\frac{1}{\frac{3}{2}+\frac{1}{2}\sqrt{5}}$$

```
> simplify(");
```

$$\frac{2}{3+\sqrt{5}}$$

```
> readlib( rationalize ): # load library function
> rationalize("");
```

$$\frac{3}{2}-\frac{1}{2}\sqrt{5}$$

```
> (-1-3*Pi-3*Pi^2-Pi^3)^(1/3);
```

$$(-1-3\pi-3\pi^2-\pi^3)^{1/3}$$

```
> simplify(");
```

$$\frac{1}{2}(\pi+1)(1+I\sqrt{3})$$

Az utolsó példában a Maple a komplex köbgyök függvény főágát használta. A valós gyök a következő trükkel kapható meg.

```
> readlib( surd ): # load library function
> surd( op(1,"""), 1/op(2,""") );
```

$$-(1+3\pi+3\pi^2+\pi^3)^{1/3}$$

```
> simplify(");
```

$$-\pi-1$$

A $\sqrt{r_1+r_2\sqrt{n}}$ alakú egymásba skatulyázott négyzetgyökök, ahol $n\in\mathbb{N}$ és $r_1,r_2\in\mathbb{Q}$, az **sqrt** és a **simplify** parancsokkal lebonthatók, ha ez lehetséges:

```
> sqrt( 4 + 2*sqrt(3) );
```

$$\sqrt{3}+1$$

```
> ( 4 + 2*3^(1/2) )^(1/2);
```

$$\sqrt{4+2\sqrt{3}}$$

```
> simplify(");
```

$$\sqrt{3}+1$$

Bonyolultabb skatulyázott radikálokat a **radnormal** eljárással egyszerűsíthetünk:

```
> readlib( radnormal ): # load library function
> sqrt(25+5*sqrt(5)) - sqrt(5+sqrt(5)) - 2*sqrt(5-sqrt(5));
```

$$\sqrt{25+5\sqrt{5}}-\sqrt{5+\sqrt{5}}-2\sqrt{5-\sqrt{5}}$$

```
> radnormal(");
```

$$0$$

A radikálok az ún. algebrai számok speciális esetei. Általánosabban az *algebrai számokon* a racionális számok fölötti egyváltozós polinomok gyökeit értjük. Így például $\sqrt{2}$ az x^2-2 polinom egyik α gyöke, $\sqrt{2}+\sqrt{3}+\sqrt{5}$ az $x^8-40x^6+352x^4-960x^2+576$ polinom egyik α gyöke, a $\sqrt{1+\sqrt{2}}$ skatulyázott radikál az x^4-2x-1 polinom egyik α gyöke. Az x^5+x+1 polinom α gyökei nem írhatók föl radikálokkal (lásd [170]-ben a megoldható ötödfokú egyenletek jellemzését). Figyeljük meg, hogy ezekben a példákban α az illető polinom *bármelyik* gyökét jelentheti, mint ahogy a 2 négyzetgyöke lehet (közelítőleg) 1.4142 vagy -1.4142.

Az algebrai számokkal végzett számítások bonyolultak és időigényesek; ez alól a Maple sem kivétel. Az algebrai számokat a „helykitöltő" **RootOf** eljárás segítségével reprezentálja a rendszer. Például

```
> alpha := RootOf( z^2 -2, z );
```

$$\alpha := \mathrm{RootOf}(_Z^2-2)$$

a 2 tetszőleges gyökét jelenti (radikál jelöléssel a $\sqrt{2}$-t vagy a $-\sqrt{2}$-t). Ebből az is kiderül, hogy a Maple az $_Z$ belső változóval számol. A **simplify** eljárás az α-t tartalmazó kifejezések egyszerűsítésénél kihasználja azt a tényt, hogy $\alpha^2=2$.

```
> simplify( alpha^2 );
```

$$2$$

```
> simplify( 1/(1+alpha) );
```

$$\mathrm{RootOf}(_Z^2-2)-1$$

Sokkal áttekinthetőbbé válnak ezek a számítások, ha a négyzetgyökre egy álnevet használunk:

```
> alias( beta = RootOf( z^2 - 2, z ) ):
> 1/(1+beta) + 1/(beta-1); simplify(");
```

$$\frac{1}{1+\beta}+\frac{1}{\beta-1}$$

$$2\beta$$

Az algebrai számok `radical`-os és `RootOf`-os reprezentációi minden lehetséges esetben könnyen átkonvertálhatók egymásba:

```
> convert( (-8)^(1/3), RootOf );
```

$$\mathrm{RootOf}(_Z^3+8)$$

```
> convert( ", radical );
```

$$(-8)^{1/3}$$

A fönti példában tehát α és β a 2 bármelyik négyzetgyöke lehet; az **allvalues** eljárás igyekszik is mindegyiküket megmutatni.

```
> allvalues( beta );
```

$$\sqrt{2}, -\sqrt{2}$$

Az **allvalues** alapföltételezés szerint valamely **RootOf** kifejezés összes előfordulását ugyanazon gyök reprezentációjának tekinti. De kérhetjük a kifejezésben szereplő **RootOf**-ok egymástól független kiértékelését is:

```
> beta + 1/beta;
```

$$\beta + \frac{1}{\beta}$$

```
> allvalues(");
```

$$\frac{3}{2}\sqrt{2}, -\frac{3}{2}\sqrt{2}$$

```
> allvalues( "", 'independent' );
```

$$\frac{3}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{2}, -\frac{3}{2}\sqrt{2}$$

A Maple-ben algebrai számtestek fölötti polinomokkal is számolhatunk. Az alábbiakban ezt a lehetőséget használjuk föl annak ellenőrzésére, hogy $\zeta = \sqrt{2} + \sqrt{3} + \sqrt{5}$ az $x^8 - 40x^6 + 352x^4 - 960x^2 + 576$ polinom gyöke és $\sqrt{2} = \frac{1}{576}\zeta^7 - \frac{7}{144}\zeta^5 - \frac{7}{72}\zeta^3 + \frac{5}{3}\zeta$. A ζ-t definiáló polinom kiszámítására rezultánsokat használunk. (V. ö. [132].)

```
> polynomial := resultant(
>   resultant( x^2-5, (x-y)^2-3, x ), (y-z)^2-2, y );
```

$$polynomial := -960\,z^2 + 352\,z^4 - 40\,z^6 + z^8 + 576$$

```
> expand( subs( z=sqrt(2)+sqrt(3)+sqrt(5), polynomial ) );
```

$$0$$

A $\sqrt{2} + \sqrt{3} + \sqrt{5}$ tehát valóban gyök. Vezessük be a ζ algebrai számot, és faktorizáljuk az $x^2 - 2$ polinomot $\mathbb{Q}(\zeta)$ fölött:

```
> alias( zeta = RootOf( polynomial, z ) ):
> factor( x^2-2, zeta );
```

$$-\frac{1}{331776}(576\,x + 960\,\zeta - 56\,\zeta^3 - 28\,\zeta^5 + \zeta^7)$$
$$(-576\,x + 960\,\zeta - 56\,\zeta^3 - 28\,\zeta^5 + \zeta^7)$$

Az $x^2 - 2$ két $\mathbb{Q}(\zeta)$ fölötti gyökét a **roots** eljárással is megkereshetjük:

```
> roots( x^2-2, zeta );
```

$$[[-\frac{1}{576}\zeta^7 - \frac{5}{3}\zeta + \frac{7}{72}\zeta^3 + \frac{7}{144}\zeta^5, 1],$$
$$[\frac{1}{576}\zeta^7 + \frac{5}{3}\zeta - \frac{7}{72}\zeta^3 - \frac{7}{144}\zeta^5, 1]]$$

A **roots** eljárás a polinomok gyökeit és multiplicitásukat számolja ki. A jelen esetben $\frac{1}{576}\zeta^7 + \frac{5}{3}\zeta - \frac{7}{72}\zeta^3 - \frac{7}{144}\zeta^5$ a 2 egy multiplicitású gyöke abban a $\mathbb{Q}(\zeta)$ testben, ahol ζ a $\zeta^8 - 40\zeta^6 + 352\zeta^4 - 960\zeta^2 + 576 = 0$ összefüggést elégíti ki.

A **RootOf** fönti leírása a polinom bármelyik gyökének manipulálására vonatkozik, anélkül, hogy megmondanánk, melyik gyökre gondoltunk. De létezik egy szelekciós mechanizmus is a gyökök kiválasztására. Néhány példa:

```
> RootOf( x^2 + 9/10, x );
```

$$\mathrm{RootOf}(10_Z^2 + 9)$$

```
> evalf(");
```

$$-.9486832981\, I$$

```
> RootOf( x^2 + 9/10, x, 1.0*I );
```

$$\mathrm{RootOf}(10_Z^2 + 9, 1.0\, I)$$

```
> evalf(");
```

$$.9486832981\, I$$

```
> RootOf( x^2 + 9/10, x, -1.0*I ..  1.0*I );
```

$$\mathrm{RootOf}(10_Z^2 + 9, -1.0\, I..1.0\, I)$$

```
> evalf(");
```

$$-.9486832981\, I$$

A **RootOf**-ban szereplő két szelektor jelentése a 2.4. táblázatban látható.

Szelektor	**A kiválasztott gyök**
`a + b*I`	az `a + b*I`-hez abszolút értékben legközelebb eső gyök
`a + b*I..c + d*I`	az **fsolve** rendezése szerinti első gyök az adott tartományból

2.4. táblázat: A **RootOf** szelektorai

2.6. Komplex számok

Az algebrai számokkal ellentétben a komplex számok a Maple alaptípusai között szerepelnek. Az i képzetes egységet ($i^2 = -1$) a Maple az `I` szimbólummal reprezentálja. A komplex számokkal végzett numerikus aritmetikai műveleteket a rendszer automatikusan végrehajtja:

```
> ( 2 + 3*I ) * ( 4 + 5*I );
```

$$-7 + 22\, I$$

```
> Re("), Im("), conjugate("), abs("), argument(");
```

$$-7,\ 22,\ -7 - 22\, I,\ \sqrt{533},\ -\arctan(\frac{22}{7}) + \pi$$

```
> 1/"";
```

$$-\frac{7}{533} - \frac{22}{533}\, I$$

A Maple sok matematikai függvényt komplex függvénynek tekint. Többértékű komplex függvények esetében a Maple a főágat használja:

```
> cos(I), ln(I), arccoth(0), sqrt(-8);
```

$$\cosh(1),\ \frac{1}{2}I\pi,\ \frac{1}{2}I\pi,\ 2I\sqrt{2}$$

```
> sqrt( (1.0+I)^2 - 1.0 );
```

$$.7861513778 + 1.272019650\,I$$

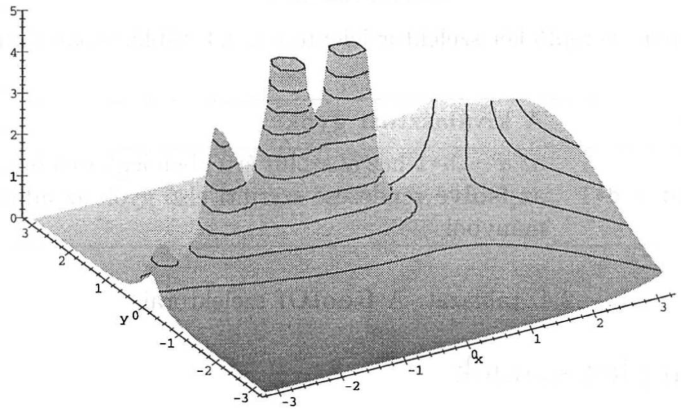
```
> Zeta( 0.5 + I ), GAMMA( 0.5 + I );
```

$$.1439364271 - .7220997435\,I,\ .3006946173 - .4249678794\,I$$

Komplex függvények abszolút értékéről közlünk két ábrát az alábbiakban:

```
> plot3d( abs( GAMMA(x+y*I) ), x=-Pi..Pi, y=-Pi..Pi,
>   view=0..5, grid=[30,30], orientation=[-120,45],
>   axes=framed, style=patchcontour );
```

A 2.8. ábrán a komplex síkon értelmezett gamma függvény abszolút értékének felületi ábrázolása látható.

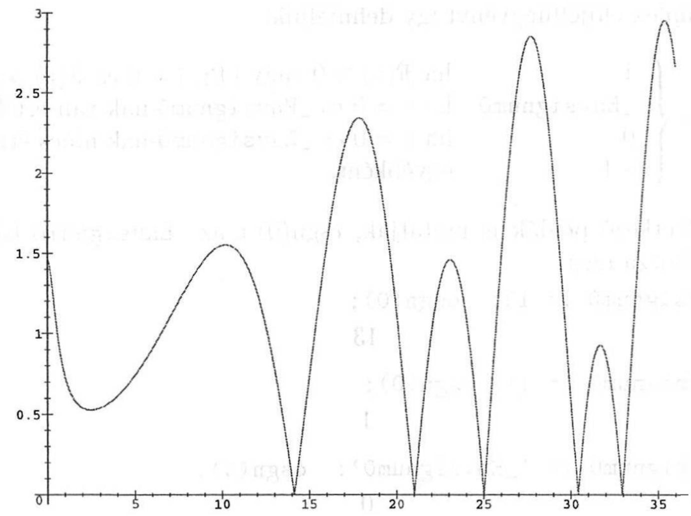


2.8. ábra: A gamma függvény abszolút értékének grafikonja

Az előző **plot3d** parancsba beírtuk a kép generálásához szükséges összes opciót. Miután a Maple kirajzolta a felületet, ezek közül sokat manuálisan megváltoztathatunk. Részleteket a grafikáról szóló 15. fejezetben találhat az Olvasó.

A 2.9. ábra a Riemann-féle zéta függvény abszolút értékét mutatja az $\Re(z) = \frac{1}{2}$ kritikus egyenesen. Látható a Riemann függvénynek erre az egyenesre eső első néhány zérushelye is. A híres Riemann hipotézis szerint minden komplex zérushely ezen a kritikus egyenesen található (lásd [107]). A Riemann hipotézist sokszor numerikusan vizsgálták; újabb keletű eredményeket tartalmaz [133, 147].

```
> plot( abs( Zeta(1/2+y*I) ), y=0..36, numpoints=1000 );
```



2.9. ábra: A Riemann-féle ζ függvény abszolút értéke a kritikus egyenesen

A Maple *szimbolikus* kifejezésekben szereplő komplex számokra vonatkozó tudásanyagát az **evalc** (= **evaluate** using **complex** number arithmetic) eljárással aktivizálhatjuk. Az **evalc** föltételezi, hogy a kifejezésben szereplő minden változó valós értékű, a komplex számokat pedig az $a + bI$ kanonikus alakba írja át, ahol a és b valós számok:

```
> 1 / (2 + p - q*I);
```

$$\frac{1}{2+p-I\,q}$$

```
> evalc(");
```

$$\frac{2+p}{(2+p)^2+q^2}+\frac{I\,q}{(2+p)^2+q^2}$$

```
> abs("");
```

$$\frac{1}{|2+p-I\,q|}$$

```
> evalc(");
```

$$\frac{1}{\sqrt{4+4p+p^2+q^2}}$$

A következő példában látni fogjuk, hogy a föltételek miként segíthetnek hozzá egyszerűbb eredmények eléréséhez bennünket.

```
> sqrt( p + q*I );
```

$$\sqrt{p+I\,q}$$

```
> evalc(");
```

$$\frac{1}{2}\sqrt{2\sqrt{p^2+q^2}+2\,p}+\frac{1}{2}\,I\,\mathrm{csgn}(q-I\,p)\,\sqrt{2\sqrt{p^2+q^2}-2\,p}$$

A **csgn** komplex előjelfüggvényt így definiáljuk:

$$\mathrm{csgn}(z)=\begin{cases}1 & \text{ha } \Re(z)>0 \text{ vagy } (\Re(z)=0 \text{ és } \Im(z)>0);\\ \texttt{_Envsignum0} & \text{ha } z=0 \text{ és } \texttt{_Envsignum0}\text{-nak van értéke;}\\ 0 & \text{ha } z=0 \text{ és } \texttt{_Envsignum0}\text{-nak nincs értéke;}\\ -1 & \text{egyébként.}\end{cases}$$

Ahogy a következő példák is mutatják, csgn(0)-t az `_Envsignum0` környezeti változó határozza meg.

```
> _Envsignum0 := 13:  csgn(0);
```

$$13$$

```
> _Envsignum0 := 1:  csgn(0);
```

$$1$$

```
> _Envsignum0 := '_Envsignum0':  csgn(0);
```

$$0$$

Miért vezették be ezt a speciális változót? És miért hívják `_Envsignum0`-nak? Az utóbbi kérdés könnyen megválaszolható: ez az ún. *környezeti változó* szabályozza `signum(0)` értékét. A **signum** eljárás természetesen a valós számokon és kifejezéseken értelmezett előjelfüggvényt jelenti, azaz

$$\mathrm{signum}(z)=\begin{cases}z/\mathrm{abs}(z) & \text{ha } z\neq 0;\\ \texttt{_Envsignum0} & \text{ha } z=0 \text{ és } \texttt{_Envsignum0}\text{-nak van értéke;}\\ 0 & \text{ha } z=0 \text{ és } \texttt{_Envsignum0}\text{-nak nincs értéke;}\end{cases}$$

A kifejezések egyszerűsítésének helyességét befolyásolhatja, hogyan definiáljuk ezt a függvényt a 0 helyen. Például a

$$\mathrm{signum}(\mathrm{abs}(x))\to 1$$

transzformáció az $x=0$ helyen hibás eredményt ad, ha $\mathrm{signum}(0)=0$. Hasonlóan, valós x és y esetén a

$$\mathrm{signum}(x\,y)\to\mathrm{signum}(x)\,\mathrm{signum}(y)$$

transzformáció sem korrekt az $x<0$ és $y=0$ esetben, ha $\mathrm{signum}(0)=1$. A Maple-nek nem kellene végrehajtani ezt az egyszerűsítést; valóban nem is végzi el automatikusan.

```
> _Envsignum0 := 1:  signum(0);
```

$$1$$

```
> signum(x*y);
```

$$\mathrm{signum}(x\,y)$$

Az `_Envsignum0` változó értéke a következőképpen szabja meg az alkalmazható transzformációkat és egyszerűsítéseket:

- Az `_Envsignum0`-nak *nincs* értéke.
  Ekkor az összes olyan transzormáció végrehajtható, ami a 0 hely kivételével mindenütt biztosan érvényes.
- Az `_Envsignum0`-nak *van* értéke.
  Ebben az esetben csak azok a transzformációk alkalmazhatók, amelyek a `signum(0)`-hoz rendelt értéket figyelembe véve is mindenütt érvényesek.

Az automatikus egyszerűsítés eléggé szövevényes terület, de látható, hogy a fölhasználó is vezérelheti bizonyos határok között.

Folytassuk az `sqrt`-vel kapcsolatos példánkat. Ha megfelelő föltételeket teszünk, a Maple tovább tud egyszerűsíteni:

```
> assume( p>0, q>0 ):
> evalc( sqrt( p + q*I ) );
```

$$\frac{1}{2}\sqrt{2\sqrt{p^{\sim 2}+q^{\sim 2}}+2\,p^{\sim}}+\frac{1}{2}\,I\,\sqrt{2\sqrt{p^{\sim 2}+q^{\sim 2}}-2\,p^{\sim}}$$

A p és a q utáni tilde karakterek azt jelzik, hogy ezekre a változókra bizonyos kikötések érvényesek. A változók tulajdonságairól a Maple **about** parancsa tájékoztat bennünket. Az **assume** részletesebb ismertetése a 13. fejezetben található.

```
> about( p );

Originally p, renamed p~:
  is assumed to be: RealRange(Open(0),infinity)
```

Befejezésül vizsgáljuk meg a -1 négyzetgyökével kapcsolatos jelöléseket. A villamosmérnökök jobban szeretnek j-t vagy J-t írni, a nagy I betű ugyanis inkább az elektromos áramkörökben folyó áram jelölésére használatos. A Maple-ben ezt így oldhatjuk meg: mivel `I` csupán az `sqrt(-1)` álneve, először ezt töröljük:

```
> alias( I = I ):
```

Azután helyette J-t veszünk:

```
> alias( J = sqrt(-1) ):
```

Innen kezdve a komplex számok standard alakja `a + bJ`:

```
> J^2;
```

$$-1$$

```
> 1/(1+J);
```

$$\frac{1}{2}-\frac{1}{2}J$$

```
> 1/(x+y*J);
```

$$\frac{1}{x+J\,y}$$

```
> evalc(");
```

$$\frac{x}{x^2+y^2} - \frac{J\,y}{x^2+y^2}$$

```
> solve( x^4 = 1, x );
```

$$1, -1, J, -J$$

Az `alias( I = I )` és az `alias( J = sqrt(-1) )` parancsok egyetlen utasításként is megadhatók: `alias( I = I, J = sqrt(-1) )`. Ha ezt beírjuk a Maple inicializáló fájlunkba, már kezdettől fogva használhatjuk a `J` jelölést.

2.7. Gyakorlatok

1. Tekintsük a következő Maple szekciót.

```
> 3^2:
> 4^2;
```

$$16$$

```
> " + "";
```

Van az utolsó utasításnak értelme? Ha igen, mi az eredménye? Ha nincs, miért?

2. Magyarázzuk meg a következő Maple parancsok eredménye közötti különbséget:

(a) `x : y;`

(b) `x/y;`

(c) `x\y;`

3. Ebben a gyakorlatban a Maple help rendszerének használatában való jártasságunkat fejleszthetjük.

(a) Tételezzük föl, hogy egy egyenlőségből, például az $1 = \cos^2 x + \sin^2 x$-ből csupán a bal vagy a jobb oldalra van szükségünk. Hogy tudjuk ezt könnyen elérni a Maple-ben?

(b) Tételezzük föl, hogy ki akarjuk számolni az exponenciális függvény lánctörtekkel való közelítését Meg tudja-e ezt a Maple csinálni? Ha igen, hajtsuk végre a számítást.

(c) Tegyük föl, hogy az $x^8 + x^6 + 10x^4 + 10x^3 + 8x^2 + 2x + 8$ polinomot modulo 13 akarjuk szorzattá alakítani. Meg tudja-e ezt a Maple csinálni? Ha igen, hajtsuk végre a szorzattá alakítást.

(d) Tételezzük fel, hogy a $\{1, 2, 3, 4, 5\}$ halmaz összes részhalmazát akarjuk meghatározni. Hogyan hajtható ez végre a Maple segítségével?

4. A `with(numtheory)` paranccsal töltsük be a `numtheory` csomagot. Néhány ismerős számelméleti függvényt is látni fogunk. A csomag rutinjai hasznosak lehetnek a következő kérdések megválaszolásánál.

(a) Konstruáljuk meg 9876543210123456789 összes osztójának listáját.

(b) Keressük meg a 9876543210123456789-hez legközelebb eső prímszámot.

(c) Mi az $5^{5^{5^5}}$ prímtényezős fölbontása?

(d) Fejtsük 10 mélységű lánctörtbe a természetes alapú logaritmus `E` alapszámát.

5. Mi a különbség az $1/3 + 1/3 + 1/3$ és az $1.0/3.0 + 1.0/3.0 + 1.0/3.0$ között Maple-ben?

6. Keressük meg $e^{\frac{1}{3}\pi\sqrt{163}}$ tíz, húsz, illetve harminc jegy pontosságú lebegőpontos közelítését.

7. Számítsuk ki π^{π^π}-t kilenc tizedesjegyre.

8. Ezt a gyakorlatot nyolc tizedesjegy pontosságú lebegőpontos számítással oldjuk meg. Mennyi a

$$310.0 \times 320.0 \times 330.0$$
$$-\sqrt{310.0 \times 320.0} \times \sqrt{320.0 \times 330.0} \times \sqrt{310.0 \times 330.0}$$

kifejezés számértéke?

9. Emlékszünk arra, hogy a $\frac{19}{6}$, a $\frac{22}{7}$ és a $\frac{25}{8}$ közül melyik adja a π elég jó racionális közelítését? Használjuk a Maple-t a legjobban közelítő szám meghatározására. Keressük meg π legjobb olyan a/b alakú racionális közelítését, ahol a és b 1000-nél kisebb természetes számok. (Útmutatás: nézzük meg a π lánctört kifejtését.)

10. Ellenőrizzük, hogy $\sqrt{2\sqrt{19549} + 286}$ egyenlő-e $\sqrt{113} + \sqrt{173}$-mal.

11. Hozzuk az $\dfrac{1}{\sqrt{3}+1}$ kifejezést $a + b\sqrt{3}$ alakra, ahol a és b racionális számok.

12. Legyen θ a $\theta^3 - \theta - 1$ polinom egyik gyöke, és vegyük a racinális számtest θ-val való kiterjesztését. Tehát $a + b\theta + c\theta^2$ alakú kifejezéseket tekintünk, ahol $a, b, c \in \mathbb{Q}$ és a kifejezésekkel való számolás során alkalmazzuk a $\theta^3 = \theta + 1$ azonosságot. Hozzuk az $\dfrac{1}{\theta^2+1}$-et a Maple segítségével $a + b\theta + c\theta^2$ alakra, ahol $a, b, c \in \mathbb{Q}$.

13. Legyen $\alpha = \sqrt{2}$, $\beta = \sqrt{3}$ és $\gamma = \sqrt{5}$. A **Primfield** eljárás segítségével számoljuk ki a $\mathbb{Q}(\alpha, \beta, \gamma)$ testbővítés egyik ζ primitív elemét, és hasonlítsuk össze az eredményt a 2.5. alfejezet utolsó példájával.

14. Mutassuk meg, hogy a Maple föl tudja írni a komplex számok e alapú hatványait a valós és a képzetes részek koszinuszának és szinuszának fölhasználásával. Számítsuk ki $e^{\pi i/12}$ értékét ebben a formában.

15. A Maple fölhasználásával igazoljuk, hogy minden $z = x + iy$ alakú komplex számra teljesül a

$$\tanh(z/2) = \frac{\sinh x + i \sinh y}{\cosh x + \cos y}$$

egyenlőség $(x, y \in \mathbb{R})$.

3. Változók és nevek

Egy Maple szekció rendszerint utasítások sorozatából áll, melyek végrehajtása során értékeket számolunk ki, majd ezeket az értékeket elnevezzük, hogy fölhasználhassuk őket későbbi számításaink során. Ebben a fejezetben azt vizsgáljuk, hogy melyek az érvényes Maple nevek, hogyan lehet egy névnek értéket adni, és hogyan lehet egy változót újra értékteleníteni, vagyis értékkel nem rendelkezővé tenni. Elmondjuk azt is, hogyan tehető védetté egy változó, hogyan szüntethető meg a védettség, végül hogyan társíthatunk változókkal attributumokat és tulajdonságokat. Az olyan programozási nyelvekkel szemben, mint a FORTRAN, az Algol és a C, a Maple-ben nem szükséges a változók típusát deklarálni. A Maple a kifejezések típusát belső ábrázolásuk, illetve használatuk alapján határozza meg. Ebben a fejezetben röviden áttekintjük az alapvető adattípusokat. Ezenkívül leírjuk a szimbolikus kifejezések szokásos kiértékelését, nevezetesen a teljes kiértékelést.

3.1. Értékadás és értéktelenítés

A számítógépes algebrai rendszerek tudományos számításokban való sikeres alkalmazásának titka abban rejlik, hogy ezekben a rendszerekben formulákkal dolgozhatunk, és megoldhatunk olyan matematikai problémákat is, melyekben ismeretlenek és paraméterek szerepelnek. A változók használatának példájaként kérdezzük meg a Maple-től az általános másodfokú egyenlet megoldóképletét. Egyenleteket általában a Maple **solve** eljárásával oldhatunk meg:

```
> solve( a*x^2 + b*x + c = 0, x );
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4\,a\,c}}{a},\ \frac{1}{2}\frac{-b-\sqrt{b^2-4\,a\,c}}{a}$$

Az a, b és c változók paraméterek, x pedig az ismeretlen, a rendszer mindegyiküket szimbólumként kezeli. A számítógépes algebrai rendszerek jellemző vonása a *szabad* vagy *nem kötött* változók használata. (Ezek a változók nem mutatnak semmilyen értékre sem a saját nevüket kivéve.)

Másrészt a változókhoz hozzáköthetünk vagy hozzárendelhetünk valamely értéket. Az *értékkel rendelkező változók* (assigned variables) használata kettős: gyakran szükségünk van a kiszámított eredmény vagy egy bonyolult kifejezés megcimkézésére a későbbi hivatkozás céljából. Amikor a Maple-t programozási nyelvként használjuk, az algoritmusainkban szereplő adatokra nevükkel hivatkozunk ahelyett, hogy közvetlenül magukat az adatokat használnánk. Ha az adat valamely rögzített speciális érték, akkor *szimbolikus konstansnak* nevezzük. A 2.4. alfejezetben már láttuk, hogy a Maple eredendően milyen szimbolikus konstansokat ismer, valamint új konstansokat fölvételének és a meglévők törlésének módjait. Ha értékkel rendelkező változókra van szükségünk, a két karakterből álló „:=" jelet (értékadási operátor) használhatjuk az értékadó utasításban. Másik lehetőség az **assign** eljárás alkalmazása, erre még visszatérünk a fejezet során:

```
> polynomial := 9*x^3 - 37*x^2 + 47*x - 19;
```

$$polynomial := 9\,x^3 - 37\,x^2 + 47\,x - 19$$

```
> roots( polynomial );
```

$$[[1, 2], [\frac{19}{9}, 1]]$$

```
> subs( x=19/9, polynomial );
```

$$0$$

```
> polynomial, x;
```

$$9\,x^3 - 37\,x^2 + 47\,x - 19,\ x$$

A fenti példában az első utasítással a `9*x^3 - 37*^2 + 47*x - 19` polinomot rendeltük értékként a `polinomial` változóhoz. Valahányszor a *polinomial* változóval találkozik a Maple, ezt az értéket veszi. Ezt a folyamatot *kiértékelésnek* nevezzük. Így a

```
roots( polinomial )
```

utasítás azzal ekvivalens, mintha a

```
roots( 9*x^3 - 37*x^2 + 47*x - 19 )
```

utasítást adtuk volna a Maple-nek. Az x változónak saját nevén kívül továbbra sincs értéke. A

```
subs( x=19/9, polinomial )
```

utasítással az `x` változót a `19/9` értékkel *helyettesítettük* a `polynomial` változóval megcimkézett polinomban, vagyis `9*x^3 - 37*x^2 + 47*x - 19`-ben. Az `x` maga azonban változatlan maradt, továbbra is önmagára mutat. Ugyanez érvényes a polinomra is, a helyettesítés hatására a `polynomial` változó nem kapott új értéket. A `19/9` gyök ellenőrzését az `x` változónak való értékadással is elvégezhetjük:

```
> x := 19/9;
```

$$x := \frac{19}{9}$$

```
> polynomial;
```

$$0$$

A Maple a fenti utasítás végrehajtásánál a következő módon jár el:

- veszi a `polimomial` változót, és `9*x^3 - 37*x^2 + 47*x - 19`-re értékeli ki,
- ezután a kifejezésben szereplő minden `x`-et `19/9`-re értékel ki,
- végül az eredményt 0-ra egyszerűsíti, tehát elvégzi a szükséges aritmetikai műveleteket.

A gyök értékadással való ellenőrzésének a helyettesítéssel szemben az a hátránya, hogy `x` most valamilyen számra mutat. Ha azt akarjuk, hogy `x` más értékre mutasson, ezt egy új értékadással érhetjük el:

```
> x := unknown:
> polynomial;
```

$$9\, unknown^3 - 37\, unknown^2 + 47\, unknown - 19$$

Ha az `unknown` változóhoz például a 7 értéket rendeljük

```
> unknown := 7;
```

$$unknown := 7$$

akkor `x` `unknown`-ra, ez pedig 7-re értkelődik ki.

```
> x;
```

$$7$$

Látható, hogy a Maple a kiértékeléseket a számítás pillanatnyi helyzetében a változókhoz rendelt értékek alapján addig végzi, amíg csak lehetséges. Ezt az általános szabályt nevezzük *teljes kiértékelésnek*:

```
> polynomial;
```

$$1584$$

Kifejezések teljes kiértékeléséről még a 3.2. alfejezetben lesz szó.

```
> x := 'x';
```

$$x := x$$

Az utolsó utasítás x-nek értékként saját nevét adja vissza. Az aposztróf karaktert használjuk arra a célra, hogy változók értékét megszüntessük. Ha egy kifejezést aposztrófok közé zárunk, a Maple nem alkalmazza a kiértékelési algoritmust a kifejezésre. Ebben a speciális esetben az x kiértékelése maradt el.

Időzzünk el még egy kicsit az indexelt és a konkatenációval képzett nevek kiértékelésénél:

```
> i := 1; A[i] := 2; A[i] := 3; B[i] := 4;
```

$$i := 1$$

$$A_1 := 2$$

$$A_1 := 3$$

$$B_1 := 4$$

Látható, hogy az := értékadó operátor baloldalán álló kifejezés egy névre értékelődik ki. Az indexelt nevek indexe kiértékelődik, de maga az indexelt név már nem: a harmadik értékadásnál a bal oldalon $A[i]$ értéke $A[1]$ lesz, de ez már nem értékelődik ki 2-re.

Az aposztrófokkal végzett értéktelenítés[1] az indexelt és a konkatenált nevekre nem működik:

```
> A[i] := 'A[i]'; # no unassignment of A[i]
```

$$A_1 := A_i$$

```
> # A[i]; but infinite recursion occurs

Error, too many levels of recursion
```

A Unix operációs rendszer esetében végtelen rekurzió föllépésekor a Maple egy bizonyos ponton a fönti hibaüzenettel abbahagyja a kritikus utasítás végrehajtását. Mást platformokon akár maga a Maple program is leállhat.

Néha a Maple figyelmeztet a rekurzív névdefiníciókra is. Ha föltesszük, hogy x szabad változó, akkor a következő utasítás az x+1 értéket rendeli hozzá:

```
> x := x+1;

Warning, recursive definition of name
```

$$x := x + 1$$

```
> # x; infinite recursion occurs

Error, too many levels of recursion
```

Az utolsó parancsban azt kértük a Maple-től, hogy értékelje ki az

$$x \to x+1 \to (x+1)+1 \to ((x+1)+1)+1 \to \cdots$$

kifejezést. A teljes kiértékelés miatt a Maple ismételten megpróbálja x-et kiértékelni, míg csak el nem éri a rekurziószámra előírt korlátot.

[1] Az „unassigment" szakkifejezés magyarításaként ezt fogjuk használni. (*A Fordító megjegyzése.*)

Az indexelt és a konkatenált nevek értéktelenítése az **evaln** (**eval**uate to a **n**ame) eljárással érhető el:

```
> A[i] := evaln( A[i] ); A[i]; # unassignment of A[i]
```

$$A_1 := A_1$$

$$A_1$$

Az **evaln** argumentumának kiértékelése szintén speciálisan történik: csak az indexet értékeli ki a Maple, és eredményként egy nevet ad vissza. Az **evaln** eljárás természetesen olyan „közönséges" változókra, mint az x, y vagy z is alkalmazható, ez az értéktelenítés „bombabiztos" módszere.

Mivel nehéz észben tartani, hogy mely változókhoz rendeltünk értéket és melyekhez nem, a számítógép-algebrai rendszer néhány segédfüggvényt biztosít ilyen célra. (Lásd a 3.1. táblázatot.)

Eljárás	**Eredmény**
anames	az értékkel bíró (kötött) változókat mutatja meg
unames	az összes érték nélküli nevet (szabad változót) sorolja föl
assigned	azt vizsgálja, hogy egy változóhoz van-e (nevétől különböző) érték hozzárendelve

3.1. táblázat: A nevek használatával kapcsolatos segédfüggvények

A következő mintapélda ezen Maple eljárások használatát illusztrálja:

```
> unames();
```

Embed, *identical*, *equation*, *anyfunc*, *endcolon*, *Factors*, *cubic*, *positive*,
................,
nonnegint, *terminal*, *unknown*, *laurent*, *anything*, *Irreduc*, *Berlekamp*,
Sprem, *Limit*, *setpath*, *Coeff*, *algnum*, *Discrim*

```
> nops( {"} ); # number of unassigned names
```

$$209$$

Az **unames** outputjának nagy részét elhagytuk, de a néhány névből, ami látható, valamint a nevek számából már világos, hogy az eljárás kevéssé használható, mivel túl részletes outputot ad. Nemcsak a fölhasználó által definiált nevek listáját, hanem a Maple által generált, de eleddig még értéket nem kapott nevek listáját is megkapjuk. Az alábbiakban a **select** paranccsal kiválasztjuk az érték nélküli változók közül a három karaktereseket:

```
> select( s -> length(s)=3, { unames() } );
```

{*all*, *row*, *not*, *and*, *set*, *Add*, *Det*, *Gcd*, *Int*, *Lcm*, *One*, *Quo*, *Rem*, *Sum*,
Svd, *odd*}

Az `s->lenght(s)=3`-hoz hasonló anoním függvények használatára a 8.8. alfejezetben visszatérünk. Másik példánk: válasszuk ki a listából az összes olyan, `con`-tól különböző nevet, amely tartalmazza a `con` részsztringet.

```
> select( s -> SearchText(con,s)>0, { unames() } )
> minus {con};
```

$$\{constant, Ratrecon, realcons\}$$

Vizsgáljuk meg részletesebben az **anames** eljárást (kezdjünk előtte új Maple szekciót).

```
> restart;
> p := q; r := q*s;
```

$$p := q$$

$$r := q\, s$$

```
> anames();
```

$$p, r$$

Az **anames** eljárás azon nevek listáját adja eredményül, amelyekhez pillanatnyilag nevüktől különböző érték van rendelve. Az előző példában ezek mind a fölhasználó által definiált változók. Ha azonban olyan neveket adunk meg, amelyekhez értékként egy bizonyos Maple típus van rendelve, akkor az output már a Maple által definiált és inicializált ilyen típusú neveket is tartalmazni fogja:

```
> anames( `*` ); # variables whose value is a product
```

$$r$$

```
> anames( name ); # variables whose value are a name
```

$$p, sysinit, integrate, mod$$

```
> anames( integer ); # variables with integer value
```

$$Digits, printlevel, Order$$

A példában előforduló fordított aposztrófok és adattípusok jelentését a 3.3. és a 3.4. alfejezetben fogjuk elmagyarázni.

Az **assigned** eljárással megkaphatjuk az egyes változók státuszát:

```
> assigned( q ), assigned( r );
```

$$false, true$$

Eddig négy kivételt láttunk a teljes kiértékelés általános szabálya alól:

- az aposztrófok közé zárt kifejezések nem értékelődnek ki,
- a `:=` értékadó operátor baloldalán álló kifejezés csak a megfelelő névre értékelődik ki, továbbá
- az **evaln** eljárás argumentuma és
- az **assigned** eljárás argumentuma

szintén csak a megfelelő nevet kapja értékül.

Ezt az alfejezetet az **assign** és az **unassign** Maple eljárások egyik lehetséges alkalmazásának bemutatásával zárjuk. A nevük sugallja funkcionalitásukat, egy ponton azonban eltérnek a korábban tanultaktól. Az

assign(name, expression)

utasítás hatása megegyezik a

name := expression

utasításával, azzal a kivétellel, hogy az **assign** eljárás első argumentuma teljesen kiértékelődik, míg ugyanez nem teljesül az értékadó operátor baloldali operanduszára. Az **assign** eljárást érdemes alkalmazni a **solve** függvény által visszaadott egyenlőségek halmazára, ha arra van szükségünk, hogy az egyes változók a megoldásként kapott értéket vegyék fel:

```
> eqns := { x + y = a, b*x - 1/3*y = c }:
> vars := { x, y }:
> sols := solve( eqns, vars );
```

$$sols := \{y = 3\,\frac{b\,a-c}{3\,b+1},\, x = \frac{3\,c+a}{3\,b+1}\}$$

```
> x, y;
```

$$x,\, y$$

```
> assign( sols );
> x, y;
```

$$\frac{3\,c+a}{3\,b+1},\, 3\,\frac{b\,a-c}{3\,b+1}$$

Ahogy a 3.6. alfejezetben látni fogjuk, az **assign** parancs jól alkalmazható (sőt szinte nélkülözhetetlen) olyan változóknak történő értékadásnál, amelyekhez bizonyos tulajdonságokat rendeltünk.

Az **unassign** eljárást fölhasználhatjuk változók egyidejű értéktelenítésére, vagy egy változó értéktelenítésére anélkül, hogy a hozzá rendelt tulajdonságokat is törölnénk.

```
> readlib( unassign ): # load library function
> unassign( 'x', 'y' ): x, y;
```

$$x,\, y$$

Könnyen kísértésbe eshetünk, hogy ezt az eljárást az **anames** paranccsal kombinálva megpróbáljuk az összes olyan változót törölni, ami eddig értéket kapott. Figyeljük meg, hogy ennek milyen drasztikus hatása lehet a Maple viselkedésére:

```
> # compute some integral
> integral := integrate( 1/(x^2+1), x );
```

$$integral := \arctan(x)$$

```
> unassign( anames() ): # unassign variables
> integrate( 1/(x^2+1), x ); # recompute the integral

Error, (in int) type ‘intargs‘ does not exist
```

A Maple hibás működésének oka az, hogy nemcsak a fölhasználó által definiált neveket, például az `integral` változót, hanem a rendszer által definiáltakat, így az `int/type` változót is értéktelenítettük. Használjuk inkább a **restart** utasítást a memória törlésére és a Maple újraindítására anélkül, hogy a rendszerből kilépnénk.
A 3.2. táblázat az értékadás és az értéktelenítés különböző lehetőségeit sorolja föl:

Minta	**A bal oldal kiértékelése**	**A jobb oldal kiértékelése**
`x := y`	névre	normális
`assign( x = y )`	normális	normális
`x := evaln( y )`	névre	névre
`x := 'y'`	névre	egy lépésben
`assign( x = evaln(y) )`	normális	névre
`assign( x = 'y' )`	normális	egy lépésben
`unassign( x )`	normális	—

3.2. táblázat: A kiértékelés módja az értékadás és az értéktelenítés során

3.2. Kiértékelés

Ha a Maple valamely utasításban egy névvel találkozik, akkor általában megkeresi azt az objektumot, amelyre az adott név mutat. Azt mondjuk, hogy a Maple *kiértékeli* a nevet. Ha a név egy másik névre mutat, akkor a rendszer azt az objektumot keresi, amelyre az újabb név mutat, és így tovább, mindaddig, amíg egy olyan objektumhoz nem ér, ami nem név, vagy ami már önmagára mutat. Az utolsó objektumot úgy használja a rendszer, mintha közvetlenül ezt helyettesítenénk az utasításba. Kifejezések esetében az általános eljárás tehát a *teljes kiértékelés*. Ez magyarázza a következő Maple szekció eredményeit:

```
> a := b; b := c; c:= 3;
```

$$a := b$$

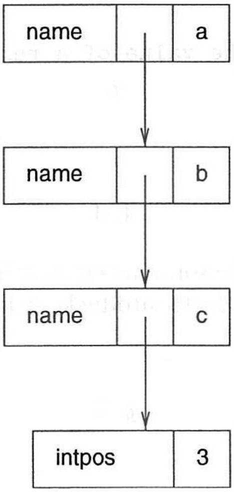
$$b := c$$

$$c := 3$$

```
> a; # evaluation of a
```

$$3$$

A teljes kiértékelés jobb megértéséhez megnézzük az értékadás hatását a belső adatstruktúrákra. A Maple a változókat háromdimenziós adatvektorokkal reprezentálja. Az első komponens jelzi, hogy a vektor egy változót reprezentál, a második komponens a változó aktuális értékére mutat, a harmadik komponens pedig magának a változónak a nevét tartalmazza. Amikor egy változónak értéket adunk, csupán a hozzárendelt értékre mutató pointert állítjuk be az illető adatvektorban. Az előző példa a 3.1. ábrán látható belső ábrázolást eredményezi. Az a változót reprezentáló adatvektor második komponense a értékére



3.1. ábra: Az értékadások utáni belső reprezentáció

mutat, tehát a értéke b. Hasonlóan látható, hogy b értéke c-vel egyenlő. Ezt a belső ábrázolást az **eval** eljárás segítségével ellenőrizhetjük:

```
> eval(a,1); # value of a
```

$$b$$

```
> eval(b,1); # value of b
```

$$c$$

```
> eval(c,1); # value of c
```

$$3$$

```
> eval(a,2); # two-level evaluation of a
```

$$c$$

```
> eval(a,3); # three-level evaluation of a
```

$$3$$

Amikor a-t teljesen kiértékeljük, az a-t reprezentáló adatvektorból indulva végigmegyünk a pointereken addig az objektumig, amelynek típusa „pozitív egész" és értéke 3. Ezt úgy mondjuk, hogy „a 3-ra értékelődik ki". Ha ezután a c változónak valamilyen új értéket, mondjuk 5-öt adunk, akkor az a változó 5-re értékelődik ki. Ugyanakkor az a változó értéke továbbra is b marad. Ezt csak értékadással változtathatjuk meg:

```
> c := 5:
> a; # a now evaluates to 5
```

$$5$$

```
> eval(a,1); # but the value of a remains the same
```

$$b$$

```
> a := 4:
> a, eval(a,1);
```

$$4, 4$$

A kifejezések teljes kiértékelése során az egyes változók a legutóbbi értékadással hozzájuk rendelt értékekkel helyettesítődnek. A következő példa ezt illusztrálja:

```
> x := y:  y := 7:
> eval(x,1), x;
```

$$y, 7$$

```
> x := x;
```

$$x := 7$$

```
> y := 9:
> x; # the value and evaluation of x remain unchanged
```

$$7$$

Az x := x utasítás azt sugallja, hogy x ezután ismét saját nevére mutat. De – mint azt már korábban láttuk – nem erről van szó! Ehhez az x := evaln(x) vagy az x: = 'x' parancsot kellett volna megadnunk. Itt az aposztrófok megakadályozzák az x teljes kiértékelését.

Amennyiben egy nevet akarunk átadni argumentumként egy Maple eljárásnak, ezt a legkönnyebben úgy érhetjük el, hogy a nevet aposztrófok közé zárjuk a teljes kiértékelés elkerülésére. Például a **rem** eljárás negyedik paraméterének, amely a hányadost tartalmazza, névnek kell lennie:

```
> #suppose that quotient has been assigned some value
> quotient := 0:
> rem( x^3 + x + 1, x^2 + x + 1, x, 'quotient' );
```

$$2 + x$$

```
> quotient; # value has been set in previous command
```

$$x - 1$$

Ha valóban biztonságra akarunk törekedni, akkor először a változónak újra saját nevét kell értékül adni a `variable := evaln(variable)` utasítással.

De mi történik, ha elfelejtjük kitenni az aposztrófokat a fenti példában?

```
> quotient := 0:
> rem( x^3 + x + 1, x^2 + x + 1, x, quotient );

Error, (in rem) Illegal use of a formal parameter
```

Mi romlott el? Az első utasítás hatására a `quotient` változó a 0 értéket kapja. A második utasítás végrehajtásakor a Maple először kiértékeli az argumentumokat, ami a `rem( x^3+x+1, x^2+x+1, x, 0 )` eredményre vezet. A Maple hibaüzenetet ad, mivel negyedik argumentumként egy nevet várt, amihez hozzárendelheti a kiszámított hányadost. Ha a `quotient` változó az x-re mutat, akkor egy végtelen rekurzív struktúra keletkezik:

```
> quotient := x:
> rem( x^3 + x + 1, x^2 + x + 1, x, quotient );
```

$$2 + x$$

```
> eval(quotient,1), eval(quotient,2);
```

$$x, x - 1$$

Tehát a belső ábrázolásban az `x := x-1` értékadás került végrehajtásra, ez pedig hibát fog eredményezni, ha később x-re hivatkozunk.

Más esetekben is kívánatos vagy éppenséggel szükséges lehet a kifejezések vagy részkifejezések teljes kiértékelésének elnyomása. Példaként számoljuk ki az első 5 prímszám összegét a **sum** eljárás segítségével:

```
> i := 0:  # assume i has been assigned some value
> sum( ithprime(i), i=1..5 );

Error, (in ithprime) argument must be a positive integer
```

A fenti függvényhívásban az `ithprime(i)` argumentum kiértékelésekor a Maple észreveszi, hogy hibás argumentumot kapott (természetes szám helyett az i-t), ezért hibaüzenetet ad. Következő próbálkozásunknál aposztrófokat alkalmazunk a túl korai kiértékelés elnyomására:

```
> sum( 'ithprime(i)', i=1..5 );

Error, (in sum) summation variable previously assigned,
 second argument evaluates to, 0 = 1 .. 5
```

Ez még mindig nem működik, de a hibaüzenet megmagyarázza a problémát. Azt is biztosítanunk kell tehát, hogy az összegzési index névre értékelődjön ki:

```
> sum( 'ithprime(i)', 'i'=1..5 );
```

$$28$$

Egy jótanács:

> *A sum-hoz hasonló utasításokban mindig tegyük aposztrófok közé az összeadandót és az összegzési indexet, kivéve, ha teljesen biztosak vagyunk benne, hogy az aposztrófok elhagyhatók.*

Aposztrófok közé zárt kifejezés kiértékelésének eredménye legjobban azzal írható le, hogy a kiértékelés egy idézőjelpár „lehámozását" jelenti.

```
> x := 1: # in this example x always has the value 1
> x+1;
```

$$2$$

```
> 'x'+1;
```

$$x+1$$

```
> 'x+1';
```

$$x+1$$

```
> '''x'+1'';
```

$$''x' + 1'$$

```
> ";
```

$$'x' + 1$$

```
> ";
```

$$x+1$$

```
> ";
```

$$2$$

3.3. Változónevek

A Maple-ben a változók és konstansok nevének legegyszerűbb fajtája a sztring (karakterlánc), vagyis olyan betűkből, számokból és alulvonásból álló sorozat, amelynek első karaktere betű vagy alulvonás. A sztringek belső ábrázolása az egész számokéhoz hasonló dinamikus adatvektor, amelyben az első szó az összes szükséges információt kódolja (a sztring típusjelzőt és a vektor hosszát), a további szavak pedig a sztring karaktereit tartalmazzák (32 bites gépen maximum négy karaktert szavanként). Hasonló a sztringekre kiszabott méretkorlátozás is: a maximális hossz $2^{19} - 9$ karakter. (V. ö. a 2.3. alfejezettel). Néhány példa:

```
> restart: # make sure that a fresh start is made
> x, a_long_name_containing_underscores, H2O;
```

$$x,\ a_long_name_containing_underscores,\ H2O$$

```
> unknown, UNKNOWN, UnKnOwN;
```

$$unknown,\ ?,\ UnKnOwN$$

Figyeljük meg, hogy a Maple megkülönbözteti a kis- és a nagybetűket, és `UNKNOWN` helyett a munkalapos interfész esetén kérdőjelet ír ki. A matematikai konstansok Maple-beli neveinek használatánál sem feledkezzünk meg erről. A π kostanst a Maple-ben a `Pi` és nem `pi` jelöli:

```
> evalf([ Pi, pi ] );
```

$$[3.141592654, \pi]$$

A görög betűk megjelenítése a munkalapokon és ezek LaTeX formátumba konvertált változatában szintén félrevezető lehet:

```
> [ Pi, pi ];
```

$$[\pi, \pi]$$

```
> latex( [ Pi, pi ] );

[\pi ,\pi]
```

Ehelyett talán a $[\Pi, \pi]$, illetve a `[\Pi,\pi]` outputot várta az Olvasó:

```
> [ Zeta, zeta ] ;
```

$$[\zeta, \zeta]$$

A $[Z, \zeta]$ szerencsésebb jelölés lenne, ezt a megoldást valószínűleg a Riemann-féle ζ-függvény Maple jelölése inspirálta.

Ügyeljünk arra is, hogy -1 komplex négyzetgyökének Maple-beli neve `I`, és nem `i`. A Maple által a természetes alapú logaritmus alapszámára használt jelölés is eléggé zavaró: nem `e`, nem is `E`, hanem `exp(1)`. A következő példából látható, mennyire óvatosaknak kell lennünk:

```
> [ e^x, exp(x) ] ;
```

$$[e^x, \mathrm{e}^x]$$

A munkalapos felületen csak a választott font különbözteti meg az outputot.

```
> latex(");

[e^{x}, e^{x}]
```

A Maple szerint valójában az első kifejezés az `e` név `x` kitevős hatványa (tehát nem a természetes alapú logaritmus alapszámának hatványa), a második pedig az `x`-re alkalmazott exponenciális leképezés. A különbség rögtön föltűnő lesz, ha behelyettesítünk egy lebegőpontos számot, és kiértékeljük az eredményt:

```
> eval(subs( x=0.5, " ));
```

$$[e^{.5}, 1.648721271]$$

and	by	do	done	elif
else	end	fi	for	from
if	in	intersect	local	minus
mod	not	od	option	options
or	proc	quit	read	save
stop	then	to	union	while

3.3. táblázat: A Maple foglalt szavai

A Maple maga is használ alulvonással kezdődő neveket (emlékezzünk vissza a 2.5. alfejezetben a **RootOf** eredményére). Vannak olyan foglalt szavak is, amelyeknek speciális jelentése van a Maple programozási nyelvében, s emiatt a rendszer közvetlenül nem fogadja el érvényes nevekként őket. A 3.3. táblázatot a **?reserved** paranccsal kaphatjuk meg.

Továbbá vannak olyan nevek is, amelyekhez már valamely jelentést társít a Maple. Ilyenek a matematikai függvények nevei (például **sin**, **cos**, **exp**, **sqrt**, ...), a Maple eljárások neve (például **copy**, **indices**, **lhs**, **rhs**, **type**, **coeff**, **degree**, **order**, ...) és Maple adattípusok neve (például **set**, **list**, **matrix**,...). Ezek többsége védett név a Maple-ben, tehát véletlenül nem adhatunk nekik új értékeket. Ha új védett neveket kívánunk fölvenni, vagy a meglévőket át akarjuk definiálni, használjuk a **protect** és az **unprotect** eljárásokat.

A következő példában az **unames** beépített eljárást terjesztjük ki oly módon, hogy outputja nevek rendezett listája legyen:

```
> unprotect(unames):
> unames := subs( old_unames=eval(unames),
> proc() sort( [old_unames()] ) end ):
> protect(unames):
> whattype( unames() ); # check whether it is a list
```

list

Az indexelt nevek védetté tétele egy kicsit speciálisan történik. Amikor egy indexelt névnek értéket adunk, egy táblát hozunk létre. A tábla neve megegyezik az illető névvel. (Lásd a 12.5. alfejezetet.) Az értékadások megelőzésére tehát az indexelt név helyett ezt kell védetté tenni:

```
> protect(A): A[1] := 2;

Error,
attempting to assign to array/table 'A' which is protected
```

A *környezeti változók* alkotják a nevek harmadik olyan csoportját, amelyekhez nem rendelhetünk tetszőleges értéket. A **?environment** parancs a beépített környezeti változók következő listáját adja:

```
Digits, Normalizer, Testzero, mod, printlevel, ", "", """
```

A `printlevel` környezeti változóval például a számolás közben a képernyőn megjelenő információk szabályozását és testreszabását végezhetjük el. Default értéke 1, de nem tehetjük fele olyan bőbeszédűvé a rendszert egy ilyen értékadással:

```
> printlevel := 1/2;

Error, printlevel must be a word size integer
```

A Maple környezeti változói a MuPAD, a Reduce vagy a Lisp rendszerekben „fluid változókként" ismertekkel analóg módon viselkednek: ezek a változók globálisan elérhetők eljárásokon belül, és az eljárás végrehajtása után automatikusan fölveszik korábbi értéküket. Egy példával megvilágítva: olyan eljárást definiálunk, amely 30 jegy pontossággal végzi el a numerikus kiértékeléseket:

```
> Digits; # the current value
                              10
> evalf30 := proc(expr)
> Digits := 30; # Digits temporarily set to 30
> evalf( expr )
> end:
> evalf30( Pi ); # Pi in 30 digits
              3.14159265358979323846264338328
> Digits; # Digits is bound to old value
                              10
```

A Maple némely környezeti változójának neve `_Env`-vel kezdődik. Ilyen például a **solve** eredményében az algebrai számok és függvények jelölésére szolgáló `_EnvExplicit` vagy a `signum(0)` értékét megadó `_EnvsignumO`. Definiálhatunk saját környezeti változókat is, ehhez csupán `_Env`-vel kezdődő neveket kell választanunk. Ügyeljünk arra, hogy a Maple is több helyen használ alulvonással kezdődő neveket. Egy jótanács:

> *Ha nem föltétlenül szükséges, kerüljük az alulvonással kezdődő neveket!*

Különben könnyen zavarba hozhatnak az alábbihoz hasonló Maple hibaüzenetek:

```
> _Z:= sqrt(2):
> RootOf(x^2-x,x);

Error, (in RootOf) expression independent of, 2^(1/2)
```

Ha az olvashatóság érdekében a neveket szóközökkel kívánjuk tagolni, vagy speciális karaktereket (pontot, kettőspontot, „/" jelet) akarunk használni, az egész nevet *fordított aposztrófok* (más néven *fordított idézőjelek*) közé kell zárni. Ezek a jelek csupán a nevek határait mutatják, ők maguk nem részei a neveknek. A sztringek megjelenítésekor a rendszer nem is mutatja őket. Két egymás után előforduló fordított aposztróf magát a fordított aposztróf karaktert jelöli. Néhány további példát mutatunk érvényes Maple nevekre:

```
> `exercise 1`, `/usr/local/lib/maple/lib/C.m`;
```

$$exercise\ 1, /usr/local/lib/maple/lib/C.m$$

```
> `Answer:`; # the colon is part of the name
```

$$Answer:$$

```
> ``; # the null string or empty name
```

```
> ```/usr/local/lib/maple/lib/C.m```;
```

$$`/usr/local/lib/maple/lib/C.m`$$

A fordított aposztrófok közé zárt Maple neveket nem úgy kell elképzelnünk, mint a hagyományos programozási nyelvek, például a FORTRAN sztringjeit. A következő példa ezt mutatja:

```
> miscellanea := `apparent_text`;
```

$$miscellanea := apparent_text$$

```
> apparent_text := cos(Pi);
```

$$apparent_text := -1$$

```
> miscellanea, `miscellanea`;
```

$$-1, -1$$

Jól látható, hogy a fordított aposztrófoknak nincs hatása. Alkalmanként meglehetősen furcsa dolgok történhetnek ezekkel az aposztrófokkal:

```
> `Here starts a name which is

Warning, string contains newline character. Close strings
with ` quote.

> concatenated` := 0;
```

$$Here\ starts\ a\ name\ which\ is\backslash$$
$$concatenated := 0$$

Ha beírunk egy fordított idézőjelet, de a további inputot elfelejtjük egy újabb fordított aposztróffal lezárni, úgy tűnik, mintha a Maple egyáltalán nem volna hajlandó inputot elfogadni. Ebből a csapdából úgy mászhatunk ki, ha követjük a Maple tanácsait: gépeljünk be egy fordított idézőjelet, utána egy pontosvesszőt, és a rendszer újra helyesen fog dolgozni.

Sokszor – különösen ha a Maple-t programozási nyelvként használjuk – jól jön a nevek konkatenációjának lehetősége. A következőkben néhány példát adunk a **cat** eljárás és a „.” konkatenációs operátor használatára.

```
> libname, cat(libname,`/C.m`);
```

$$/usr/local/maple/update, /usr/local/maple/lib,$$
$$/usr/local/maple/update/usr/local/maple/lib/C.m$$

```
> `` . `/usr/local/lib/maple/lib` . `/` . `C.m` ;
```

$$/usr/local/lib/maple/lib/C.m$$

régebben „." helyett „!!" volt!?


```
> X.Y, X.1;
```

$$XY, X1$$

```
> X . ( Y, 1 );
```

$$XY, X1$$

```
> X . (1..8);
```

$$X1, X2, X3, X4, X5, X6, X7, X8$$

```
> “ . ( X, Y ) . ( 1 .. 4 );
```

$$X1, X2, X3, X4, Y1, Y2, Y3, Y4$$

```
> i:=4: X.i, i.X;
```

$$X4, iX$$

Az utolsó példa azt mutatja, hogy ha a konkatenáció műveletét használjuk, akkor az első név nem értékelődik ki. A tervezők célja nyilvánvalóan az volt, hogy az `x.1` például mindig `x1`-re értékelődjön ki, függetlenül attól, hogy az x-hez rendeltünk-e valamilyen értéket. Néhány esetben ezért használtunk a fönti példában üres nevet a konkatenáció művelet első argumentumaként.

Ne keverjük össze a különböző idézőjelek használatát a Maple-ben! Hatásukat a 3.4. táblázatban foglaljuk össze.

Idézőjel	**Alkalmazásának célja**
‘ ‘	speciális karaktereket tartalmazó név jelzése
’ ’	kiértékelés elnyomása
"	hivatkozás az előzőleg kiértékelt kifejezésre
""	hivatkozás az utolsó előtt kiértékelt kifejezésre
"""	hivatkozás a hárommal korábban kiértékelt kifejezésre

3.4. táblázat: Különféle idézőjelek

3.4. Alapvető adattípusok

Az adatok lehetséges értékeit osztályokra, úgynevezett *adattípusokra* bontjuk. Az adattípus többek között azt rögzíti, hogy az adatok milyen értékeket vehetnek föl, vagy hogy milyen műveleteket végezhetünk rajtuk. A szokásos elemi adattípusok, például az *integer* (egész szám), *floating point number* (lebegőpontos szám) és a *string* (karakterlánc) mind megtalálhatók a Maple-ben, de ezeken kívül még sok más adattípus is létezik. A Maple-ben az adatok alaptípusát a **whattype** eljárással kérdezhetjük le.

```
> whattype( 5.0 );
```

float

```
> whattype( ‘an example of a long name‘ );
```

string

```
> whattype( { 1, 2, 3 } );
```

set

```
> whattype( 1, 2, 3 );
```

exprseq

Adat	**Adattípus**	**Példa**
Számok és sztringek		
egész	`integer`	`1`
tört	`fraction`	`1/2`
lebegőpontos szám	`float`	`0.33333`
alfanumerikus szöveg	`string`	`xvalue`
Aritmetikai kifejezések		
összeg	`‘+‘`	`x + y`
szorzat	`‘*‘`	`x * y`
hatvány	`‘^‘` vagy `‘**‘`	`x ^ y`
Relációs kifejezések		
egyenlet	`‘=‘, equation`	`x+1 = 1+x`
egyenlőtlenség	`‘<>‘`	`Pi <> pi`
kisebb reláció	`<`	`2 < 3`
kisebb-egyenlő reláció	`‘<=‘`	`E <= Pi`
Logikai kifejezések		
és-kifejezés	`‘and‘`	`P and Q`
vagy-kifejezés	`‘or‘`	`P or Q`
negáció	`‘not‘`	`not P`
Összetett kifejezések		
kifejezés-sorozat	`exprseq`	`a,b,c`
halmaz	`set`	`{a,b,c}`
lista	`list`	`[a,b,c]`
tábla	`table`	`table[a,b,c]`
indexelt név	`indexed`	`X[1]`
függvényhívás	`function`	`f(x), itt f definiálatlan név`
Egyéb adatok		
kiértékeletlen konkatenáció	`‘.‘`	`a.(1..n), itt n-nek nincs értéke`
tartomány	`‘..‘, range`	`1 .. 3`
általánosított hatványsor	`series`	x^{-1} `- gamma + O(x)`
eljárásdefiníció	`procedure`	`proc(x) x^3 end`
kiértékeletlen kifejezés	`uneval`	`”x + y”`

3.5. táblázat: Gyakori felületi adattípusok

A **whattype** eredményül az adatvektor fejrészének leírását adja. Azokat az adattípusokat, melyekhez elegendő az adatvektor fejrészét ismerni, felületi adattípusoknak nevezzük. A 3.5. táblázatban felsoroljuk a Maple leggyakoribb fe-

lületi adattípusait. A **?surface** parancs segítségével megkaphatjuk ezek teljes listáját is.

A Maple további három eszközt kínál a típusok tesztelésére, nevezetesen a **type**, a **hastype** és a **typematch** eljárásokat:

```
> type( x + 1, `+` );
```

true

```
> hastype( x + 1, `+` );
```

true

```
> typematch( x + 1, (a::string) &+ (b::integer) );
```

true

```
> a, b;
```

$x, 1$

```
> hastype( x + 1/2 * y, 'fraction' );
```

true

```
> hastype( x + 2 * y, 'fraction' );
```

false

Ez a három típustesztelő függvény nemcsak a felületi adattípusokat teszteli, hanem kifejezések általánosabb típusait is vizsgálja. Bejárják a kifejezéshez rendelt fát, és az úgynevezett *skatulyázott adattípusokat* (nested data types) is tesztelik. Használhatók még *strukturált adattípusok* vizsgálatára is. Ezek olyan nevektől különböző kifejezések, amelyek adattípusokként interpretálhatók. Néhány értelmes skatulyázott és strukturált adattípus:

```
> type( x^2 + x + 1, polynom );
```

true

```
> type( { x^2 + x + 1, x^2 - x }, set(polynom) );
```

true

```
> type( x^2 + x + Pi, polynom(integer,x) );
```

false

```
> type( x^2 + x + 1, 'quadratic'( x ) );
```

true

Az utolsó eljáráshívás második argumentumában az aposztrófokat azért használtuk, hogy elnyomjuk az argumentum kiértékelését, ami gondot okozna abban a nem túl valószínű esetben, ha a `quadratic` névhez már valamilyen értéket rendeltünk volna. Célszerű megszokni, hogy a **type** második argumentumát mindig aposztrófok közé tesszük. Végül is nem minden, a Maple által használt belső név védett. Strukturált típusok esetén ez a megoldás már kényelmetlenné válhat.

Parancs	A tesztelt típus
whattype	felületi adattípus
type	skatulyázott vagy strukturált adattípus
hastype	bizonyos típusú részkifejezés(ek) létezése
typematch	strukturált típus tesztelése és összehasonlítása

3.6. táblázat: Típusok tesztelése

A 3.6. táblázatban összefoglaltuk, hogyan lehet típusokat tesztelni a Maple-ben. A **typematch** a legerősebb parancs: ha egy strukturált típusnak megfelelő egyezéseket talál, akkor az összes mintaváltozóhoz a sikeres mintaillesztésnek megfelelő értéket rendel. Például

```
> typematch( exp(x), a::exp(b::name) );
```

$$true$$

```
> a, b;
```

$$e^x, x$$

Figyeljük meg a dupla kettőspont (::) használatát a **typematch** parancs második részében. Ez nem két utasítás-elválasztó, hanem egy olyan operátor, amely egy szimbólumot egy adattípussal párosít. A 3.7. táblázatban összefoglaljuk a nevek, valamint a kettőspontot, pontosvesszőt és egyenlőségjelet tartalmazó operátorok használatát.

Operátor	Alkalmazása
:	utasítás-elválasztó
;	utasítás-elválasztó
=	egyenlet vagy egyenlőség reláció
::	típus vagy tulajdonság hozzárendelése
:=	értékadás

3.7. táblázat: A kettőspont, a pontosvessző és az egyenlőségjel kombinációi

A hagyományos programozási nyelvekben, például a FORTRAN-ban a változók típusát deklarálnunk kell, és ez nem változtatható meg a programon belül. Egészen más a helyzet a Maple-ben: itt a változóknak nincs rögzített típusa. Máskülönben az interaktív munkát túlzottan megterhelővé tennék a típusdeklarációk, melyek majdnem mindig fölöslegesek, hiszen a matematikai kontextusból

világos, hogy milyen értékeket vehetnek föl a változók. A típuskonverziók is nehézkessé válnának. Néhány illusztratív példa:

```
> number := 1: whattype( number );
```

integer

```
> number := 0.75: whattype( number );
```

float

```
> number := convert( number, fraction );
```

$$number := \frac{3}{4}$$

```
> convert( number, 'binary' );

Error, (in convert/binary) invalid argument for convert
```

Az utolsó utasítás azt szemlélteti, hogy nem minden típusváltás megengedett; a konverziónak a rendszer számára értelmesnek kell lennie. Például a $\frac{3}{4}$ racionális számot fölírhatjuk a következő lánctörtként (**con**tinued **frac**tion):

$$0 + \cfrac{1}{1 + \cfrac{1}{3}}$$

```
> convert( number, 'confrac' );
```

$$[0, 1, 3]$$

Az aposztrófok a **convert** eljárás második argumentuma körül ismét azt a célt szolgálják, hogy elkerüljük a változó kiértékelését. A `confrac` nem védett név a Maple-ben! A `type( name, protected )` paranccsal tudhatjuk meg, hogy a `name` név védett-e.

A számelméleti célokat szolgáló `numtheory` csomag **numtheory**[**cfrac**] eljárása olvashatóbb kétdimenziós outputot ad:

```
> # load the function cfrac from the numtheory package
> with( numtheory, cfrac ):
> cfrac( number );
```

$$\cfrac{1}{1 + \cfrac{1}{3}}$$

A Maple a *p-adikus számokkal* való számoláshoz tartalmaz egy `padic` nevű csomagot is. Írjuk föl például a $\frac{3}{4}$ törtet triadikus szám alakjában:

```
> with( padic ): # load the p-adic number package
> evalp( number, 3 );
```

$$3 + 2\,3^2 + 2\,3^4 + 2\,3^6 + 2\,3^8 + O(3^{10})$$

3.5. Attributumok

A Maple konstans méretű adatstruktúrái (ilyenek a *name, list, set, procedure, function* és a *float*) egy *attributumnak* nevezett extra információs mezővel is rendelkezhetnek. Attributumként bármely érvényes Maple kifejezés megengedett. Az attributumokat a **setattribute** és az **attributes** függvényeken kívül egyetlen más Maple függvény sem látja, és nem is képes módosítani. A **setattribute** eljárás attributummal rendelkező adatstruktúrát hoz létre. az **attributes** egy struktúra attributumait adja vissza. A Maple kernel fölismer bizonyos nevekhez kapcsolódó attributumokat, például a `protected`-et. Néhány példa:

```
> setattribute( A, `capital A` );
```

$$A$$

```
> attributes(A);
```

$$capital\ A$$

```
> attributes( diff );
```

$$protected$$

```
> setattribute( differentiate, protected );
```

$$differentiate$$

```
> differentiate := diff;
Error, attempting to assign to `differentiate` which is protected
> S := setattribute( {2,3,5,7,11}, `first 5 primes` );
```

$$S := \{2, 3, 5, 7, 11\}$$

```
> attributes( S );
```

$$first\ 5\ primes$$

```
> S := setattribute( algebraicset(
> [ x*y*z^2 - 4, x^2 +y^2 +z^2 - 1, x - y - 1 ],
> [ x, y, z ] ), dimension = 0, groebnerbasis =
> [ x - y - 1, 2*y^2 + z^2 + 2*y, 8 + z^4 ] );
```

$$S := \mathrm{algebraicset}([x\,y\,z^2 - 4,\, x^2 + y^2 + z^2 - 1,\, x - y - 1],\, [x,\, y,\, z])$$

```
> attributes( S );
```

$$dimension = 0,\ groebnerbasis = [x - y - 1,\, 2\,y^2 + z^2 + 2\,y,\, 8 + z^4]$$

3.6. Tulajdonságok

Az **assume** és az **additionally** eljárásokkal rendelhetünk változókhoz tulajdonságokat. A Maple megpróbálja számolás közben fölhasználni ezeket a tulajdonságokat, és minden tőle telhetőt megtesz, hogy az adott tulajdonságokból kiindulva levezesse a kifejezések tulajdonságait. Most csak néhány bevezető példát mutatunk, és a megjelenítéssel, valamint az értékadással kapcsolatos bizonyos technikai kérdéseket tisztázunk. A részletesebb tárgyalásra a 13. fejezetben kerül sor.

```
> (-1)^(m^2+n); # no assumptions
```

$$(-1)^{(m^2+n)}$$

```
> assume( m, odd ):
> assume( n, odd ):
> [ (-1)^m, (-1)^(m+n), cos(m*Pi)];
```

$$[(-1)^m, (-1)^{(m+n)}, -1]$$
with assumptions on m and n

```
> simplify(");
```

$$[-1, 1, -1]$$

Látható, hogy a Maple teljesen magától „megérti”, hogy a π páratlan egész számú többszörösének koszinusza 1. A **simplify** végrehajtásakor azt is fölhasználja, hogy páratlan szám négyzete ismét páratlan, valamint két páratlan egész szám összege páros. Mindebből helyes következtetéseket von le a -1 hatványokra vonatkozóan.

A változókkal kapcsolatos tulajdonságok az **about** eljárással vizsgálhatók:

```
> about( m );

Originally m, renamed m~:
  is assumed to be: LinearProp(2,integer,1)
```

A rendszer tehát egy tilde (~) jel hozzáfűzésével átnevezi azokat a változókat, amelyekhez valamely tulajdonságot rendeltünk. Nézzük meg ezt az `m` változóra, és az **additionally** eljárással adjunk neki még egy tulajdonságot:

```
> m;
```

$$m\tilde{}$$

```
> additionally( m>0 ):
> about( m );

Originally m, renamed m~:  is assumed to be:
AndProp(LinearProp(2,integer,1),
                    RealRange(Open(0),infinity))
```

Az $m > 0$ tulajdonság megadása tehát az „m eleme a nyitott $(0, \infty)$ intervallumnak” tulajdonság hozzáadását eredményezte. De változott-e maga a név? Próbáljuk ki:

```
> m - ";
```

$$m\tilde{} - m\tilde{}$$

Micsoda meglepetés! A titok nyitja az, hogy a Maple minden egyes föltételhez egy új lokális nevet generál, ami az értékadásoknál is bizonyos következményekkel jár.

```
> m := 3:
> about( m );

3:
  All numeric values are properties as well as objects.
  Their location in the property lattice is obvious,
  in this case integer.

> m := odd_number:
> about( m );

odd_number:
  nothing known about this object
```

Ha tehát értéket adunk egy változónak, minden hozzá rendelt tulajdonság törlődik.

```
> assume( n, integer );
> expr := cos(n*Pi);
```

$$expr := (-1)^{n\tilde{}}$$

```
> n := 2:
> expr;
```

$$(-1)^{n\tilde{}}$$

Vajon mi történt? Az n változónak 2-t adtunk értékül, ez tényleg egy egész szám. Az értékadás hatására törlődnek az n-hez korábban rendelt tulajdonságok. Az expr kifejezésben azonban még mindig az n~-ra átnevezett régi változóra hivatkozunk. Ennek elkerülésére az egyik megoldás az **assign** használata, mivel ez kiértékeli argumentumait. Szükség esetén az **eval**-t használhatjuk a kiértékelés mélységének vezérlésére. Egy példa:

```
> assume( N, integer );
> expr := N;
```

$$expr := N\tilde{}$$

```
> assign( N = 2 ):
> expr;
```

$$2$$

```
> assign( eval(N,1), 3 );
> expr;
```

$$3$$

A fejezet végén három olyan trükköt ismertetünk, amelyekkel az outputból eltüntethető a hozzárendelt tulajdonságokkal rendelkező változók nevének végén álló tilde jel. Az első trükk az **alias** eljárást használja:

```
> assume( p>0 ):
> p;
```

$$p^{\sim}$$

```
> alias( 'p' = p ):
> p;
```

$$p$$

```
> about( p );

Originally p, renamed p~:
  is assumed to be: RealRange(Open(0),infinity)
```

Ez csak „optikai csalódás". A **latex** parancs például továbbra is kiírja a tildét.

```
> latex( p );

\mbox {{\tt `p~`}}
```

A Maple V Release 4 `showassumed` nevű interfész változója jobb módszert kínál a tildéktől való megszabadulásra. Értéke 0, 1 vagy 2 lehet. A 0 beállítás letiltja a föltételekkel bíró változók jelölését. Az alapföltételezés szerinti 1 értéknél a nevek végén az eddig látott tilde jelenik meg. Végül 2-t értékül adva a kifejezések végén jelenik meg azon változók listája, amelyekhez valamely föltételt kapcsoltunk:

```
> interface( showassumed = 0 );
> assume( q>0 );
> q;
```

$$q$$

```
> about( q );

Originally q, renamed q~:
  is assumed to be: RealRange(Open(0),infinity)

> latex( q );

\mbox {{\tt `q~`}}

> interface( showassumed = 2 );
> assume( r>0 );
> r;
```

$$r$$

with assumptions on r

```
> about( r );

Originally r, renamed r~:
  is assumed to be: RealRange(Open(0),infinity)
```

```
> latex( r );
```

```
\mbox {{\tt `r~`}}
```

Még jobb módszer a tildéktől való megszabadulásra a Maple fölülbírálása. Először nézzük meg a tildék megjelenítését végző kódot:

```
> readlib( assume ): # load assume from library
> interface( verboseproc = 3 ): # make Maple communicative
> print( `property/Rename` ); # print responsible routine
```

```
proc(nm)
local nam;
global _AName;
option `Copyright (c) 1992 Gaston Gonnet, \
Wissenschaftliches Rechnen, ETH Zurich. All rights reserved.`;
    if assigned(`property/OrigName`[nm]) then
        nam := eval(`property/OrigName`[nm], 1)
    else nam := nm
    fi;
    subs('DUMMY2' = ``.nam.`~`,
        proc(nam)
        local DUMMY2;
        global `property/OrigName`;
            `property/OrigName`[DUMMY2] := nam; nam := DUMMY2
        end );
    nm = "(nam)
end
```

Most már könnyű a kódot úgy átalakítani, hogy a tilde ne jelenjen meg:

```
> `property/Rename` := proc(nm)
>   local nam;
>   global _AName;
>   if assigned(`property/OrigName`[nm]) then
>        nam := eval(`property/OrigName`[nm], 1)
>   else nam := nm
>   fi;
>   subs('DUMMY2' = ``.nam,
>     proc(nam)
>       local DUMMY2;
>       global `property/OrigName`;
>       `property/OrigName`[DUMMY2] := nam;
>       nam := DUMMY2
>     end
>   );
>   nm = "(nam)
> end:
> assume( s>0 ):  s; # no tilde!
```

$$s$$

```
> about( s );
```

```
Originally s, renamed s~:
  is assumed to be: RealRange(Open(0),infinity)

> latex( s );

\mbox {{\tt `s~`}}
```

3.7. Gyakorlatok

1. Mi a különbség az a és a b változó használatában az alábbi két Maple parancsban?

```
> a := 'a';  b := 'b';  a := b;  b := 10;
```

és

```
> a := 'a'; b := 'b'; b := 10; a := b;
```

Kiegészíthetnénk-e az alábbi parancssort egy további értékadással úgy, hogy annak a végső hatása megegyezzen a legelső utasítássorozattal?

```
> a := 'a'; b := 'b'; b := 10;
```

2. Részletesen magyarázzuk el, hogy mi történik, ha végrehajtjuk a következő utasítássorozatot.

```
> p := 'p' ; q := 'q' ;
> 7 * 5 ;
> p = 10 ;
> q : 3 ;
> "" ; "" ; "" ;
> p ; q ;
```

3. Magyarázzuk meg a következő Maple parancsok különböző eredményét. `x*y;`, `x.y;`, és `x\y;`.

4. Írjuk le részletesen a következő két Maple szekció közötti különbséget (például a belső adatstruktúrák fölrajzolásával).

```
> a := b ;
> b := 3 ;
> a;
> b := 4 ;
> a;
```

és

```
> b := 3 ;
> a := b ;
> a;
> b := 4 ;
> a;
```

5. Próbáljuk megjósolni a következő utasítások eredményét, és ellenőrizzük válaszunkat a Maple segítségével.

```
> X1 := 5; j := 1;
> X.j;
> 'X.j';
> 'X'.j;
> `X.j`;
```

6. Tételezzük föl, hogy v1, v2 és v3 értékkel rendelkezik. Miért nem lehet értékteleníteni ezeket a változókat a következő do-ciklussal?

```
> for i to 3 do v.i := 'v.i' od:
```

Keressünk két másmilyen ciklust, amelyek a kívánt megoldást adják.

7. Magyarázzuk meg, hogy miért hibásak a következő parancsok.

```
> gcd( x^2 - 1, x -1 , x );
> x;
```

8. Hajtsuk végre a következő parancsokat, és jósoljuk meg eredményüket, majd vessük össze a rendszer válaszaival.

```
i := 3: x:= 4:
sum( x^i, i = 1..5 );
sum( x^i, 'i' = 1..5 );
sum( 'x^i', 'i' = 1..5 );
sum( 'x'^i, 'i' = 1..5 );
sum( x^'i', 'i' = 1..5 );
sum( 'x'^`i`, 'i' = 1..5 );
sum( `x^i`, 'i' = 1..5 );
'sum( x^i, i = 1..5 )';
```

9. Az $a + b + c$ kifejezést hozzuk fokozatosan az $a * b * c$ alakra, ezt pedig az $[a, b, c]$ formára.

10. A Maple segítségével határozzuk meg $\sqrt{2}$-nek, $e - 1$-nek és az aranymetszés $\frac{1+\sqrt{5}}{2}$ arányának lánctört kifejtését.

4.

Ismerkedés a Maple rendszerrel

Ez a fejezet részletesen ismerteti, hogy a Maple hogyan kezeli az input és az output műveleteket, hogyan alakíthatjuk át a rendszert saját ízlésünknek megfelelően (a prompt jel és a nyomtatási szélesség beállítása, cimkék használata stb.), hogyan szerkeszthetjük az input adatokat, hogyan olvashatunk és írhatunk fájlokat, hogyan nyerhetünk több információt a program által fölhasznált számítógépes erőforrásokról, hogyan követhetjük nyomon számításainkat, valamint hogyan kaphatunk információkat a kiválasztott technikákról és algoritmusokról. A formázott I/O és a kódgenerálás a Maple és más programozási vagy szövegformázó nyelvek kapcsolatát példázza. Ezen felül megmagyarázzuk a Maple könyvtár felépítését (standard könyvtár, vegyes könyvtár, csomagok és az osztott könyvtár).

A fejezetben tárgyalt témák némelyike függ a fölhasználói felülettől és/vagy az operációs rendszertől. Föltételezzük, hogy az Olvasó Unix-kompatibilis gépen a munkalapos fölhasználói felülettel dolgozik. Alkalmanként arra is kitérünk, mindez hogyan működik karakteres fölhasználói felületen vagy nem-Unix platformokon.

4.1. Input és output

Egy begépelt parancsra a Maple különféle módon reagálhat:

- A rendszer értelmezi a parancsot, és a következő válaszok valamelyikét adja:
  - vagy néhány soros választ kapunk, amelyet egy új prompt jel követ,

– vagy azonnal egy prompt jelet kapunk, jelezve, hogy a Maple egy újabb parancsra vár.

Az első esetben az input sort pontosvesszővel zártuk le, a második reakció akkor fordulhat elő, ha kettőspontot használtunk. A kettőspont használata esetén a Maple a számítást „csöndben" végzi el. A fejezet további részében majd néhány kivételt is felsorolunk ez alól az általános szabály alól.

- Amennyiben az utasításunkat még nem fejeztük be, például elfelejtettük kitenni a parancsokat elválasztó pontosvesszőt vagy a kettőspontot, akkor a Maple ismét csak nem válaszol, hanem egy új prompt jelet ad. Ekkor még mindig befejezhetjük az utasításunkat:

```
> Digits := 25: evalf(
> log(cos(Pi/12)) )
> ;
```

$$-.03466823209753695510470884$$

- Ha a Maple nem érti az általunk beírt utasítást, akkor szintaktikus hibaüzenetet („Syntax error") kapunk. A karakteres interfész esetében a Maple a ^ („caret") szimbólummal jelzi azt a helyet, ahol a szintaktikus hibát fölfedezte. Munkalapok esetén ehelyett villogó kurzor jelenik meg az adott helyen. Ebben az esetben csak azt tehetjük, hogy kijavítjuk a hibás sort, és újra megadjuk az utasítást az általunk helyesnek vélt formában:

```
> x*
> *3;

Syntax error, `*` unexpected

> x**
> 3;
```

$$x^3$$

Ha a munkalapos fölhasználói felületet használjuk, az input korrigálásához a kurzort bárhova állíthatjuk (legtöbb esetben azért, hogy az elfelejtett zárójeleket kitegyük, vagy hogy beszúrjunk egy $*$-ot vagy más műveleti jeleket stb.), és újra végrehajtathatjuk a parancsot. Amennyiben a helyettesítéses mód („replace mode") opció van bekapcsolva, az újra generált output helyettesíti az előzőt. A karakteres interfész egy beépített sorszerkesztőt („line editor") tartalmaz, amely az utoljára megadott száz input sor szerkesztését engedi meg.

- Bár az utasításunk szintaktikailag helyes, a Maple rendszer mégis megtagadja annak végrehajtását:

```
> read a_non_existing_file;
```

```
Error, unable to read `a\_non\_existing\_file`
```

```
> 123456789 ^ 987654321;
```

```
Error, object too large
```

Egy hibaüzenet után a rendszer általában kész az új input fogadására. Méginkább igaz ez, ha a hibajelzés után egy üres utasítást írunk be egy önmagában álló pontosvessző formájában. Más a helyzet, hogyha például egy bal idézőjellel kezdődő nevet elfelejtettünk lezárni egy újabb bal idézőjellel. Ebben az esetben előbb be kell zárnunk a nevet bal idézőjellel, s csak ezután várja majd a Maple az új parancsot:

```
> `This is a long name;
> 2+3; @%!
> that contains funny characters`;
```

This is a long name; \
2 + *3*; @%!\
that contains funny characters

A Maple outputban a „\" karakter a folytatósorokat jelzi.

- A # („hashmark") szimbólum használatával megjegyzés sort adtunk a rendszernek. A Maple nem viszhangozza a # utáni szöveget:

```
> a := 2 # a := two ;
> ;
```

$$a := 2$$

```
> a;
```

$$2$$

- Bár megváltoztathatók a Maple által éppen végrehajtandó számítások bizonyos föltételei, a rendszer sokszor nem veszi figyelembe ezeket a változtatásokat, hanem csupán egy előzőleg kapott eredményt vesz elő valamely belső táblázatából. Erre egy példa:

```
> eqns := { seq( x[i]^2 = x[i], i=1..7 ) };
```

$$eqns := \{ x_1{}^2 = x_1, x_2{}^2 = x_2, x_3{}^2 = x_3, x_4{}^2 = x_4, x_5{}^2 = x_5, x_6{}^2 = x_6, x_7{}^2 = x_7 \}$$

```
> nops( { isolve(eqns) } );
```

$$100$$

Valójában $2^7 = 128$ egész megoldás létezik. A `_MaxSols` változó vezérli a visszaadott megoldások maximális számát. Mégis, ha ezen változó értékét megnöveljük, és újra meghívjuk az **isolve** függvényt, a Maple csupán előveszi az előző eredményt, és ugyanúgy 100-at ad vissza.

```
> _MaxSols := 500:
> nops( { isolve(eqns) } );
```

100

A helyes válaszhoz először a **forget** függvénnyel rá kell venni a rendszert, hogy felejtse el az előző eredményt:

```
> readlib( forget ): # load library function
> forget( isolve ): # forget previous results of isolve
> nops( { isolve(eqns) } ); # all solutions found
```

128

A Maple reakciója tehát függ attól, hogy milyen módon fejeztük be parancsunkat, illetve hogy az aktuális szekcióban eddig mi történt. Ennél azonban több befolyást is gyakorolhatunk a keletkező outputra. Először is létezik egy `printlevel` nevű változó, amelynek az alapértelmezése 1. Ha ezt a változót negatívra állítjuk, akkor semmiféle eredményt nem ír ki a rendszer (függetlenül attól, hogy a parancsot pontosvesszővel vagy kettősponttal zártuk le). Másrészt viszont minél nagyobb pozitív egész értéket adunk ennek a változónak, a Maple rendszer többet árul el arról, hogy min is dolgozik éppen. A 2 és 4 közé eső értékek azt a célt szolgálják, hogy magyarázó megjegyzéseket kapjunk a kiválasztott algoritmusról, illetve információkat a paraméterekről, lokális változókról, vagy futási idő alatti („run time") hiba előfordulása esetén az éppen végrehajtott utasításról.

Minél magasabb értéket adunk a `printlevel` változónak, annál több információt kapunk. Ha egy Maple program tesztelését („debug") végezzük, egyáltalán nem szokatlanok a 100 és 1000 közti nagy értékek sem. (Bár hasonló célokra használhatjuk a rendszer beépített debuggerét is, lásd **?debugger**.)

Ha csak arra vagyunk kíváncsiak, hogy milyen technikák vagy algoritmusok kerülnek végrehajtásra, akkor a **userinfo** opcióit használhatjuk, ezeket az `infolevel`[*function*]:= *level* vagy az `infolevel`[*all*]:= *level* utasítással állíthatjuk be. A 4.1. táblázat tartalmazza az egyes szinteken a rendszer által a fölhasználónak nyújtott információkat:

Szint	**Eredmény**
1	minden szükséges információ
2,3	általános információk, beleértve a fölhasznált technikákat és algoritmusokat
4,5	részletes információk a probléma megoldásáról

4.1. táblázat: A **userinfo** lehetőségei

Két példát mutatunk a `printlevel` és az `infolevel` használatára:

```
> printlevel := 2: # set printlevel a higher value
> lhs( x + y );

Error, (in lhs) invalid arguments
```

```
executing statement: ERROR(`invalid arguments`)

lhs called with arguments: x+y
```

Ha a printlevel-t 2-re állítjuk, a Maple nem lesz az átlagosnál sokkal bőbeszédűbb, kivéve, ha valami hiba következik be; ekkor arról is informál bennünket, hogy mi volt a meghívott függvény, mik voltak az argumentumok, és mennyi volt a lokális változók értéke. Ezek az adatok némileg rávilágíthatnak arra, hogy miért nem elvárásaink szerint alakultak a dolgok:

```
> printlevel := 1: # reset printlevel
> infolevel[ integrate ] := 1: # raise infolevel
> integrate( 1/(x^3+x+1), x = 0 .. infinity );

int/indef:   first-stage indefinite integration
int/ratpoly:   rational function integration
int/rischnorm:   enter Risch-Norman integrator
int/risch:   enter Risch integration
int/risch:   exit Risch integration
```

$$-\left(\sum_{_R=\%1} _R \ln(-\frac{62}{9}_R^2 + \frac{31}{9}_R + \frac{4}{9})\right)$$
$$\%1 := \mathrm{RootOf}(31_Z^3 - 3_Z - 1)$$

Ha a printlevel változót negatív értékre állítottuk be, de azért látni akarjuk számításunk eredményeit a terminál képernyőjén, akkor erről külön gondoskodnunk kell. Erre használhatjuk a **print** vagy az **lprint** eljárások valamelyikét. A különbség mindössze a megjelenítés módjában mutatkozik meg: a **print** eljárás úgynevezett kétdimenziós megjelenítést produkál, amely a szokásos matematikai jelölésekre emlékeztet, már amennyire ez lehetséges. Az **lprint** eljárás („linear print") az eredményt egydimenziós formában, balra igazítva jeleníti meg. Nézzünk egy példát:

```
> printlevel := -1: # Maple becomes silent
> sols := solve( a*x^2 + b*x + c, x );
> print( sols ); # 2-dimensional output
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4\,a\,c}}{a},\ \frac{1}{2}\frac{-b-\sqrt{b^2-4\,a\,c}}{a}$$

```
> lprint( sols ); # left-adjusted, 1-dimensional output

1/2/a*(-b+(b^2-4*a*c)^(1/2))     1/2/a*(-b-(b^2-4*a*c)^(1/2))

> printlevel := 10: # Maple gives more details
> integrate( 1/ ( x^5 + 1 ), x = 0 .. infinity );

{--> enter int, args = 1/(x^5+1), x = 0 .. infinity
```

$$_Envsignum0 := _Envsignum0$$
$$x := x$$
$$ra := 0$$

$$rb := \infty$$

$$indef := false$$

$$_EnvIndefinite := false$$

$$g := \frac{1}{x^5+1}$$

$$surds := false$$

$$r := \frac{1}{25}\pi\sqrt{5}\sqrt{2}\sqrt{5+\sqrt{5}}$$

```
<-- exit int (now at top level) = 1/25*Pi*5^(1/2)*2^(1/2)*(
5+5^(1/2))^(1/2)}
```

$$\frac{1}{25}\pi\sqrt{5}\sqrt{2}\sqrt{5+\sqrt{5}}$$

```
> printlevel := 1:  # reset printlevel to default.
```

Egyébként a **print** és az **lpint** eljárások eredményére nem hivatkozhatunk sem a " operátorral, sem más módon, mivel mindkét eljárás csak mellékhatásként végzi el a formulák megjelenítését, de valójában mindkettő a speciális `NULL` értéket adja vissza. Ezt a speciális nevet használja a Maple rendszer az üres kifejezés-sorozat és az üres utasítás jelölésére is. Ha a " dupla aposztróffal hivatkozunk az előző eredményre, a `NULL` értéket a rendszer átugorja. Figyeljük meg a különbséget a következő utasításokban:

```
> Example := 11/12, evalf(exp(-10)), NULL, (1-x)/(1+x):
> `some text`;
```

some text

```
> lprint( Example ); # display of some expressions

11/12   .4539992976e-4   (1-x)/(1+x)

> ";
```

some text

```
> Example:  # evaluation to a sequence of expressions
> ";
```

$$\frac{11}{12}, .00004539992976, \frac{1-x}{1+x}$$

```
> interface( prettyprint = 0 );
> ";

11/12   .4539992976e-4   (1-x)/(1+x)
```

Az utolsó két utasítás azt mutatja, hogy a `prettyprint` nevű interfész változót nullára állítva a Maple outputja balra igazított egydimenziós formátumú lesz. A 4.6. fejezetben látni fogjuk, hogyan állíthatjuk be további interfész opciók segítségével saját ízlésünk szerint a Maple rendszert.

4.2. A Maple könyvtár

A Maple könyvtár négy részből áll:

- a standard könyvtár,
- a vegyes könyvtár („miscellaneuos library”),
- a csomagok és
- az osztott könyvtár („share library”).

A Maple könyvtár nagysága az osztott könyvtárat nem számítva kb. 10 Mb. A rendszer azonban csak azokat a függvényeket tölti be a memóriába, amelyekre aktuálisan szüksége van.

Valahányszor egy olyan utasítást adunk ki, amely olyan standard könyvtárbeli eljárást igényel, amelyet korábban még nem használtunk a Maple szekció során, a rendszer elég okos ahhoz, hogy a kívánt eljárást a külső tárolóról a memóriába töltse, majd végrehajtsa a számítást.

A vegyes könyvtár ritkábban használt eljárásokat tartalmaz. Ezeket az eljárásokat használat előtt a **readlib** utasítással a külső tárolóból a memóriába be kell tölteni:

```
> f := exp(a*z) / ( 1 + exp(z) ); # some formula
```

$$f := \frac{e^{(a\,z)}}{1+e^z}$$

```
> residue( f, z=Pi*I );
```

$$\mathrm{residue}(\frac{e^{(a\,z)}}{1+e^z},\, z = I\,\pi)$$

A vegyes könyvtárban található a **residue** eljárás, amely egy komplex függvény valamely pólusa körüli reziduumát számítja ki. Ezt eddig még nem töltöttük be a memóriába. Ezért a Maple az inputot változatlanul hagyja, és megvárja, míg definiáljuk, vagyis a könyvtárból a memóriába betöltjük a függvényt.

```
> readlib( residue ); # load library function
```

$$\mathbf{proc}(f,\, a) \ldots \mathbf{end}$$

```
> residue( f, z=Pi*I );
```

$$-e^{(I\,a\,\pi)}$$

```
> residue( 1 / ( z^2 + a^2 ), z=a*I );
```

$$-\frac{1}{2}\frac{I}{a}$$

A Maple speciálisabb célokat szolgáló csomagokat is rendelkezésünkre bocsát. Az `orthopoly` csomag az ortogonális polinomokat (többek között a Hermite, Legendre, Laguerre és a Chebyshev polinomokat) tartalmazza. Ezeknek a függvényeknek a fölhasználásakor meg kell adnunk a csomag nevét is. Például egy Csebisev-polinom kiszámításához a következő parancsot kell kiadnunk:

```
> orthopoly[T](4,x);
```

$$8\,x^4 - 8\,x^2 + 1$$

Négy lehetőségünk van arra, hogy elkerüljük a hosszú nevek használatát.

- Az **alias** eljárás segítségével szinonímákat definiálhatunk:

```
> alias( T = orthopoly[T] ):
> T(5,x);
```

$$16\,x^5 - 20\,x^3 + 5\,x$$

A `T` definíció törléséhez a következő utasítást kell kiadnunk:

```
> alias( T = T ):
```

- A **macro** eljárás használata, melynek segítségével rövidítéseket definiálhatunk:

```
> macro( T = orthopoly[T] ):
> T(6,x);
```

$$32\,x^6 - 48\,x^4 + 18\,x^2 - 1$$

A **macro** és az **alias** eljárás közti különbség abban áll, hogy a macro szolgáltatás egy egyszerű rövidítési mechanizmus, amely csak az input adatokat változtatja meg,

```
> 'T(6,x)';
```

$$\mathit{orthopoly}_T(6,\,x)$$

míg az **alias** lehetőséget ad mind az input, mind az output oldal befolyásolására. (Lásd még a 2.5. fejezet példáit.) A makrodefiníció törléséhez a következő utasítást kell kiadnunk:

```
> macro( T = T ):
```

- A `T` Csebisev-polinom explicit betöltése az `orthopoly` csomagból:

```
> with( orthopoly, T );
```

$$[T]$$

```
> T(7,y);
```

$$64\,y^7 - 112\,y^5 + 56\,y^3 - 7\,y$$

- Annak közlése a Maple-lel, hogy az egész csomagot használni akarjuk:

```
> with( orthopoly );
```

$$[G,\,H,\,L,\,P,\,T,\,U]$$

A **with(orthopoly)** utasítás hívása azt eredményezi, hogy nevek egy csoportja (pl. a **T** vagy az **U**) ugyanazon eljárásokra mutat, mint a csomagok esetében szokásos, nekik megfelelő (**orthopoly[T]** ill. **orthopoly[U]**) hosszú elnevezések. Az eljárások teljes kódja *nem* töltődik be a könyvtárból:

```
> T(8,z);
```

$$128\,z^8 - 256\,z^6 + 160\,z^4 - 32\,z^2 + 1$$

Valójában csak ezen a ponton töltődik be és hajtódik végre a megadott argumentumokkal az **orthopoly[T]** eljárás kódja.

A **?index,packages** parancs begépelésével megkaphatjuk a Maple könyvtárban rendelkezésünkre álló csomagok teljes listáját. A csomagok tartalmazhatnak „részcsomagokat" is. Ilyen fölépítésű például a `stats` csomag, amelynek függvényei a 4.2. táblázatban látható hét részcsomagba vannak csoportosítva:

Részcsomag	**Fölhasználás**
`anova`	variancia-analízis
`describe`	adat-analízist végző függvények
`fit`	lineáris regresszió
`transform`	adatmanipulációs függvények
`random`	véletlenszám-generátorok
`statevals`	eloszlások numerikus kiértékelése
`statplots`	statisztikai rajzoló függvények

4.2. táblázat: A `stats` csomag részcsomagjai

Numerikus értékek listájának (számtani) közepe például a `describe` részcsomag **mean** eljárásával számítható ki. Ez az eljárás többféleképpen is aktivizálható:

```
> data := [ seq(i,i=1..6) ];
```

$$data := [1, 2, 3, 4, 5, 6]$$

```
> stats[describe,mean](data); # direct call
```

$$\frac{7}{2}$$

```
> with( stats ): # load stats package
> describe[mean]( data ); # semidirect call
```

$$\frac{7}{2}$$

```
> with( describe ): # load subpackage
> mean( data ); # call after loading subpackage
```

$$\frac{7}{2}$$

Végül létezik egy osztott könyvtár (`share library`), amely a fölhasználók által fejlesztett Maple eljárásokat és csomagokat tartalmazza. A **?share,address** parancs megadásával megtudhatjuk, hogy honnan kaphatjuk meg az osztott könyvtár legfrissebb verzióját, a **?share,contrib** parancs pedig azt írja le, hogyan kapcsolódhatunk be magunk is a könyvtár fejlesztésébe. Ha a Maple rendszert helyesen installáltuk, akkor az osztott könyvtárban lévő kódot a következő paranccsal érhetjük el:

```
> with(share):
```

Innentől kezdve az osztott könyvtár összes eljárását vagy csomagját ugyanúgy olvashatjuk vagy tölthetjük be, mintha közönséges Maple eljárások vagy csomagok lennének. Például betölthetjük a Ritt–Wu-féle karakterisztikus halmaz

módszer Dongming Wang-féle implementációját (lásd [186]), és segítségével algebrai egyenletrendszereket oldhatunk meg.

```
> readshare(charsets,algebra):
> with( charsets );
```

$$[cfactor, charser, charset, csolve, ecs, eics, ics, iniset, ivd, mcharset, mcs, mecs, qics, remset, triser]$$

```
> polys := [x^2 - 2*x*z + 5, x*y^2 + y*z^3,
> 3*y^2 - 8*z^3 ];
```

$$polys := [x^2 - 2\,x\,z + 5, x\,y^2 + y\,z^3, 3\,y^2 - 8\,z^3]$$

```
> vars := [ x, y, z ]:
> sols := csolve( polys, vars );
```

$$sols := \{y = 0, x = -I\sqrt{5}, z = 0\}, \{y = 0, x = I\sqrt{5}, z = 0\},$$
$$\{x = \%1, y = -\frac{8}{3}\%1, z = \frac{1}{2}\frac{\%1^2 + 5}{\%1}\}$$
$$\%1 := \mathrm{RootOf}(3_Z^6 - 64_Z^5 + 45_Z^4 + 225_Z^2 + 375)$$

4.3. Fájlok írása és olvasása

A Maple rendszerben a következő fájltípusokat különböztetjük meg:

- munkalap fájlok („worksheets");
- fájlok, amelyek Maple kódot tartalmaznak olyan formában, amelyet csak a számítógépes algebrai rendszer képes értelmezni;
- fájlok, amelyek a fölhasználó által olvasható és értelmezhető Maple kódot tartalmaznak, tehát például olyanok, amelyek a Maple programozási nyelvén íródtak;
- fájlok, amelyek Maple számítások eredményeit tartalmazzák, de önmagukban nem használhatók további számítások inputjaként; és végül
- fájlok, amelyek olyan formázott inputot/outputot tartalmaznak, amelyeket más fölhasználói programok készítettek, vagy amelyeket más programok olvashatnak be inputként.

A fájlok legutolsó kategóriájával a későbbi fejezetekben foglalkozunk. Figyelmünket most csak az első négy fájltípusra koncentráljuk.

Először tekintsük a munkalap fájlokat. Szokásos kiterjesztésük `.mws` (Maple WorkSheet). Ezek olyan ASCII fájlok, amelyek a munkalapok input, output és grafika celláin kívül a formázáshoz szükséges stílus-információt is tartalmaznak. Betölthetjük őket újabb Maple szekciókba, és újra végrehajtathatjuk a bennük szereplő parancsokat.

Ezután foglalkozzunk a negyedik csoporttal, a számítási eredmények tárolására vagy megjelenítésére szolgáló fájlokkal. Normális esetben a pontosvesszővel befejeződő Maple parancsok eredményei beágyazódnak a munkalapba, illetve megjelennek a képernyőn. De fájlba is átirányíthatjuk az outputot annak érdekében, hogy aztán nyugodtan újra elolvashassuk, vagy éppenséggel kinyomtassuk az eredményeket. Például ahhoz, hogy egy Maple szekció összes eredményét átirányítsuk az `outputfile` nevű fájlba, elegendő a következő parancs megadása:

```
> writeto( 'outputfile' );
```

Innentől kezdve az összes Maple output a munkalap vagy a terminál képernyője helyett ebbe a fájlba kerül. Itt azonban némi eltérés van a munkalapos és a karakteres alapú fölhasználói felület között. Az utóbbi esetében a „>" prompt jel, a kiadott utasítások és a futásidőben keletkező üzenetek is beíródnak a fájlba. Ennek oka az, hogy a karakteres alapú fölhasználói felület esetén a Maple nem képes egyidejűleg írni a fájlba és a terminál képernyőjére. A begépelt utasítások sorban végrehajtódnak. Tegyük föl, hogy a következő utasításokat adtuk ki:

```
> number := 90;
> polynomial := x^2 + 81;
> subs( x =number, polynomial );
> writeto( terminal );
```

Az utolsó utasítás után ismét látható lesz az összes output a terminál képernyőjén:

```
> ";
```

$$8181$$

```
> quit
```

Az `outputfile` nevű fájl a következőképpen néz ki (föltéve, hogy a karakteres alapú fölhasználói felület esetén előzőleg az **interface(echo = 0)** paranccsal letiltottuk az input parancsok és a memóriahasználat kiíratását):

```
                    number := 90

                                   2
                polynomial := x  + 81

                      8181
```

Ne felejtsük el, hogy a **writeto** a létező fájlokat felülírja. Ha ezt el akarjuk kerülni, vagy éppenhogy egy már meglevő fájlunkhoz akarjuk hozzáfűzni az outputot, akkor az **appendto** eljárást kell használnunk. Ahogy már korábban megjegyeztük, az output fájllal az olvasáson és a nyomtatáson kívül nem sok mindent tehetünk. További feldolgozás nélkül nem tudjuk fölhasználni egy újabb Maple szekció inputjaként. A **save** eljárás segítségével a Maple számítások eredményei úgy menthetők el, hogy később egy másik Maple szekcióban valóban újra fölhasználhatók legyenek. Egy ilyen fájlból a Maple **read** parancsával olvashatjuk

be az adatokat a memóriába. Az output fájlba az adatok átvitele kétféle módon történhet meg:

- csak a Maple számára érthető „belső formátumban" (ha „`.m`" kiterjesztésű fájlnevet adtunk meg),
- „emberi fogyasztásra alkalmas", simán olvasható szövegformátumban.

Nézzünk egy példát:

```
> restart; # ensure a fresh start of maple
> polynomial := x^2 + 2*x +1; `number four` := 4;
```

$$polynomial := x^2 + 2\,x + 1$$

$$number\ four := 4$$

```
> save datafile;
> save `datafile.m`;
> quit
```

Ne felejtsük el kitenni a fordított aposztrófokat a `datafile.m` név köré, különben a Maple a `datafilem` névvé konkatenálja a pont bal és jobboldalán lévő sztringet, és ennek következtében az eredményt nem a belső, hanem a fölhasználó által olvasható szöveges formában írja ki a fájlba. Megjegyezzük, hogy akkor is szükséges a fordított aposztróf használata, ha alkönyvtárakat akarunk megadni, és szükségünk van a „/" Unix-szeparátorra, hiszen enélkül a rendszer osztásjelként értelmezné. Ha Unix operációs rendszerű gépen dolgozunk, a home könyvtár rövid jelölése, „~" sem használható, ehelyett ki kell írni a teljes elérési utat. Ennek hiányában a rendszer a fájlok írásánál és olvasásánál az elérési utakat ahhoz a könyvtárhoz viszonyítva értelmezi, ahonnan a Maple-t elindítottuk.

A `datafile` és a `datafile.m` fájlok mindegyike az összes olyan információt tartalmazza, amely a Maple szekcióban az adatátvitel előtt rendelkezésre állt. Mindez lehet egy új Maple szekció kiindulási pontja. A `datafile` fájlt elolvashatjuk (és kívánságaink szerint módosíthatjuk is) a **cat** vagy **more** Unix parancsokkal, illetve valamely szövegszerkesztővel. A Maple program futása közben „belülről" ezek a parancsok az **ssystem** eljárással érhetők el. Ez két értékből álló listát ad vissza: az első elem a végrehajtott Unix parancs visszatérési értéke, a második egy Maple név, amely a parancs összes outputját tartalmazza. Csak ezt a második elemet fogjuk tekintetbe venni:

```
> ssystem( `cat datafile` )[2];

                    polynomial := x^2+2*x+1;
                        `number four` := 4;
```

A Maple által választott „egydimenziós elrendezés" miatt az előző sorok olvashatósága hagy még némi kívánnivalót maga után. De lássuk mi történik, ha ezt a fájlt beolvassuk egy új Maple szekcióban!

```
> read datafile;
```

$$polynomial := x^2 + 2\,x + 1$$

$$number\ four := 4$$

```
> `number four` * polynomial;
```

$$4x^2 + 8x + 4$$

```
> quit
```

Láthatjuk, hogy a `datafile` fájlban levő utasításokat úgy kezeli a rendszer, mintha közvetlenül a billentyűzetről adtuk volna be őket. Ebből következik, hogy a **read** eljárás számára kedvenc szövegszerkesztőnk segítségével úgy készíthetünk elő egy alkalmas fájlt, hogy Maple utasításokat írunk egymás után.

A Maple rendszer indításkor először mindig egy, a fenti módon beolvasható inicializáló fájlt keres. Például a UNIX operációs rendszer alatt a Maple először egy `init` nevű rendszerszintű inicializáló fájlt keres a Maple könyvtár `src` alkönyvtárában. Ezután a fölhasználó home könyvtárában a `.mapleinit` iniciálizációs fájlt olvassa be. Ezeknek a fájloknak a fölhasználásával automatikusan betölthetünk olyan csomagokat, amelyekre mindig szükségünk van, beállíthatunk interfész változókat (lásd a 4.7. fejezetet), és így tovább.

Amikor a **read** eljárás az összes adatot betöltötte a fájlból, a Maple az előző input folyamból érkező utasítások feldolgozásához tér vissza: ezek érkezhetnek egy munkalapból, a terminálról vagy egy másik fájlból.

A **read** utasítás hatására keletkező összes bejövő adat újra megjelenik a képernyőn. Néha jól jön ez az emlékeztető, de más esetekben túl sok információt jelenthet. Az adatok megjelenítését elnyomhatjuk azáltal is, hogy két új sort fűzünk a fájl elé, melyben megőrizzük a `printlevel` változó aktuális értékét, majd -1-re állítjuk be; a fájl végéhez pedig hozzáfűzzük azt az utasítást, amely visszaállítja a `printlevel` eredeti értékét. Mindez persze extra munka. A későbbiek során látni fogjuk, hogy a fájlból érkező input echozása nem történik meg, ha az adatokat belső Maple formátumban tároljuk. A belső formátum használatának legfontosabb előnye éppen az, hogy az olvasás és az írás sokkal hatékonyabban történik, mint az olvasható formátum esetén. Az alábbiakban használt `datafile.m` nem a fölhasználó által olvasható, hanem belső Maple formátumú. Ez a fájl is ASCII karakterekből áll, így például e-maillel is könnyen átvihető:

```
> read `datafile.m`;
> polynomial := factor( polynomial );
```

$$polynomial := (x+1)^2$$

```
> squarenumber := subs( x=`number four`, polynomial );
```

$$squarenumber := 25$$

Nem kötelező az összes változót elmenteni, választhatunk közülük. Az előző szekció folytatásaként a `number four` változót és az $(x+1)^2$ kifejezést elmentjük az `output` nevű olvasható fájlba:

```
> save `number four`, polynomial, output;
```

Az output fájl tartalma a következő lesz:

```
'number four' := 4;
polinomial := (x+1)^2;
```

A Maple belső formátumában tárolt fájlok gyors beolvasása és a rendszer szűkszavúsága még fontosabbá válik, amikor a Maple könyvtárból töltjük be az eljárásokat. Valószínűleg nem akarjuk látni minden egyes könyvtári függvény betöltésekor a használatával vagy definíciójával kapcsolatos üzeneteket.

De miért szükséges egy külön eljárás, nevezetesen a **readlib** a könyvtári függvények beolvasásához? Miért nem használjuk a **read**-et? A válasz a Maple által feltételezett hierachikus fájlstruktúrában rejlik. A fájlokba való írás és olvasás művelete mindig arra az alkönyvtárra vonatkozóan történik meg, amely az aktuális könyvtár volt, amikor a Maple-t elindítottuk (kivéve, ha teljes elérési utat adtunk meg). A Maple könyvtár teljes elérési útját a `libname` változó értékének a Maple szekción belüli kiíratásával kaphatjuk meg. Az eredmény az alábbihoz hasonló lesz:

```
> libname;
```

$$/usr/local/maple/update,\ /usr/local/maple/lib$$

Innen az `mtaylor.m` fájlt például a kövekező parancsokkal tölthetjük be:

```
> read `/usr/local/maple/lib/mtaylor.m` :
```

vagy

```
> read ``. libname . `/mtaylor.m` :
```

A Maple könyvtár eljárásainak ilyen elérési módja a hosszú nevek használata miatt kényelmetlen és könnyen elhibázható. Sokkal kellemesebb ez a rövid alak:

```
> readlib( mtaylor ):
```

Úgy tűnik, mintha a Maple ezt az utasítást a következőképpen értelmezné:

```
> read `` . libname . `/` . mtaylor . `.m`:
```

Valójában ennél többet tesz: ellenőrzi, hogy az **mtaylor** nevű eljárás tényleg definiálva lesz-e a beolvasás eredményeként; és ha igen, akkor azt adja vissza, mint a **readlib** utasítás értékét. Ebből következik, hogy rögtön argumentumokat is megadhatunk, és egy többváltozós Taylor-sort egyetlen paranccsal is kiszámíthatunk a következő módon:

```
> readlib( mtaylor ):
> read `` . libname . `/` . mtaylor . `.m`:
> readlib( mtaylor )( sin(x+y), [x,y] );
```

$$x+y-\frac{1}{6}x^3-\frac{1}{2}y\,x^2-\frac{1}{2}y^2\,x-\frac{1}{6}y^3+\frac{1}{120}x^5+\frac{1}{24}y\,x^4+\frac{1}{12}y^2\,x^3+\frac{1}{12}y^3\,x^2 \\ +\frac{1}{24}y^4\,x+\frac{1}{120}y^5$$

```
> sort( simplify( ", {z=x+y}, [x,y,z] ) );
```

$$\frac{1}{120}z^5-\frac{1}{6}z^3+z$$

```
> subs( z=x+y, " );
```

$$\frac{1}{120}(x+y)^5 - \frac{1}{6}(x+y)^3 + x + y$$

Szerencsére a legtöbb Maple-eljárás hívásakor automatikusan betöltődik a könyvtárból. Ilyen *readlib-definiált függvény* a **gcd** is; ennek eredeti definíciója `gcd := 'readlib('gcd')'`.

```
> eval( gcd, 1 ); # value of gcd
> gcd( x^3-1, x^2-1 );
```

readlib('*gcd*')

$$x - 1$$

A readlib-definiált függvények tehát automatikusan betöltődnek hívásuk pillanatában. Így a fölhasználó szempontjából nem is különböztethetők meg a Maple kerneljében definiált függvényektől.

A Maple rendszerben egynél több könyvtárat is használhatunk. Például ha betöltjük az *osztott könyvtárat*, akkor a `sharename` nevű változó az osztott könyvtár elérési útját tartalmazza, a `libname` változó értéke pedig a két könyvtár nevéből álló lista lesz. A könyvtárakban a keresést a listán látható sorrendben végzi a rendszer:

```
> with( share ):
> sharename;
```

/usr/local/maple/share

```
> libname;
```

/usr/local/maple/lib, */usr/local/maple/share*

Több könyvtár egyidejű használatának támogatása lehetőséget ad a fölhasználónak arra, hogy saját könyvtárral is rendelkezzen, vagy akár felülírja a Maple könyvtárat. A következő paranccsal arra utasítjuk a Maple rendszert, hogy először keressen a saját könyvtárunkban, és csak ha ez nem vezet eredményre, akkor nyúljon a standard könyvtárhoz minden olyan esetben, amikor a **readlib** eljárást vagy valamely readlib-definiált függvényt hívunk:

```
> libname := `/ufs/heck/private_maplelib`, libname:
```

A 4.3. táblázat a Maple fájlok írásával és olvasásával kapcsolatos parancsokat összegzi:

4.4. Numerikus adatok importja és exportja

Numerikus adatok importja

A Maple **readline** és **readdata** eljárásai tetszőleges (formázatlan) adatok fájlból vagy a terminálról való bevitelére szolgáló segédeszközök. A **readline** egy sort, a **readdata** oszlopokba rendezett numerikus adatokat olvas be a megadott helyről. Tegyük föl, hogy az aktuális könyvtárban rendelkezésünkre áll a következő 3 soros, `numericaldata` nevű fájl:

Parancs	Jelentése
read(*file*)	Maple input beolvasása az olvasható formátumú *file* fájlból
read(*'file.m'*)	Maple input beolvasása a belső formátumú *file* fájlból
readlib(*parancs*)	*parancs* beolvasása a megfelelő könyvtár-fájlból
save(*file*)	az összes változó értékének elmentése az olvasható formátumú *file* fájlba
save(*'file.m'*)	az összes változó értékének elmentése a belső formátumú *file* fájlba
save(*változósorozat, file*)	a fölsorolt változók értékének elmentése az olvasható formátumú *file* fájlba
appendto(*file*)	a további eredmények hozzáfűzése az olvasható formátumú *file* fájlhoz
writeto(*file*)	a további eredmények kiírása az olvasható formátumú *file* fájlba
writeto(*terminal*)	a további eredmények kiírása a terminálra

4.3. táblázat: Maple fájlok írása és olvasása

```
1       0.84     .54
2       0.91    -.42
3       0.14    -.99
```

Az alábbiakban néhány példát mutatunk arra, hogyan olvasható be a teljes fájl, illetve bizonyos részei.

```
> readlib( readdata ): # load the library function
> readdata( 'numericaldata', 3 ); # read 3 columns of floats
```

$$[[1., .84, .54], [2., .91, -.42], [3., .14, -.99]]$$

```
> readdata( 'numericaldata', 2 ); # read 2 columns of floats
```

$$[[1., .84], [2., .91], [3., .14]]$$

```
> readdata( 'numericaldata' ); # read 1st column of floats
```

$$[1., 2., 3.]$$

A **readdata** oszlopokba rendezett, üreshely-karakterekkel elválasztott egészeket vagy valós számokat fogad el. Ha egészként kívánjuk beolvasni az adatokat, ezt külön meg kell adni.

```
> readdata( 'numericaldata', integer );
```

$$[1, 2, 3]$$

```
> readdata( 'numericaldata', [integer,float,float] );
```

$$[[1, .84, .54], [2, .91, -.42], [3, .14, -.99]]$$

```
> readline( 'numericaldata' ); # read first line
```

$$1\ 0.84\ .54$$

```
> readline( 'numericaldata' ); # read second line
```

$$2\ 0.91\ -.42$$

Az utóbbi két kifejezés `string` típusú:

```
> whattype(""), whattype(");
```

$$string, string$$

Az adatfájlok sokszor vegyesen tartalmaznak számokat és szöveget. Az ilyen bonyolultabb fölépítésű adatfájlok csak formázott inputként olvashatók be; erre jó a **readline** és a **fscanf** eljárás. A részletekre a következő alfejezetben még visszatérünk, most csupán azt mutatjuk meg, hogyan bontható szét az utoljára kapott sztring az **sscanf** alkalmazásával egy egész és két valós számra:

```
> sscanf( "", '%d%f%f' );
```

$$[2, .91, -.42]$$

Numerikus adatok exportja

A **writeline** és **writedata** eljárások formázatlan adatok fájlba vagy a terminálra való kiírására szolgáló segédeszközök. A **writeline** a megadott Maple kifejezéseket újsor karakterekkel elválasztva írja ki a megadott fájlba, a fájlt egy újsor karakterrel zárja le. A **writedata** numerikus adatokat ír fájlba vagy a terminál képernyőjére.

Tegyük föl, hogy a következő adatokat tartalmazó mátrixot definiáltuk:

```
> data := matrix( 3, 3, [ [1, 0.84, 0.54],
> [2, 0.91, -0.42],
> [3, 0.14, -0.99] ] );
```

$$data := \begin{bmatrix} 1 & .84 & .54 \\ 2 & .91 & -.42 \\ 3 & .14 & -.99 \end{bmatrix}$$

A következő parancs ezeket a numerikus adatokat lebegőpontos számokként jeleníti meg a képernyőn:

```
> writedata( terminal, data, float );

1       .84     .54
2       .91     -.42
3       .14     -.99
```

Ezeket az adatokat kiírathatjuk például az `outputfile` nevű fájlba:

```
> writedata( outputfile, data, float );
```

A fájl tartalma az alábbi módon jeleníthető meg:

```
> ssystem( 'cat outputfile' )[2];

        1 .84 .54
        2 .91 -.42
        3 .14 -.99
```

Részletesebben is specifikálhatjuk az adatok formátumát:

```
> writedata( terminal, data, [integer,float,float] );

1    .84   .54
2    .91   -.42
3    .14   -.99
```

A **writedata** képes nemnumerikus (azaz nem egész, racionális vagy valós) adattípusok kezelésére is. Sőt, egy Maple eljárás megadásával az általunk választott formátumban történik a kiírás. Egy tipikus példa: a nemnumerikus értékek helyett írassunk ki egy csillagot:

```
> for i to 3 do data[i,1] := evaln( data[i,1] ) od:
> print(data);
```

$$\begin{bmatrix} data_{1,1} & .84 & .54 \\ data_{2,1} & .91 & -.42 \\ data_{3,1} & .14 & -.99 \end{bmatrix}$$

```
> writedata( terminal, data, float,
> proc(x) printf('*',x) end );

*    .84   .54
*    .91   -.42
*    .14   -.99
```

4.5. Alacsony szintű I/O

A Maple íráskor és olvasáskor megengedi általános objektumok, ún. folyamok[1] használatát. A folyam input vagy output forrása lehet. Folyamokra példák a fájlok, a Unix csövek, a terminál képernyője.

Alapvetően a következő alacsony szintű séma szerint írhatunk folyamokba outputot. Először outputra megnyitjuk a folyamot, vagyis tájékoztatjuk a Maple-t, hogy outputot akarunk küldeni az illető fájlba vagy csőbe. A megnyitás után írhatunk a folyamba. Végül a folyamot lezárjuk.

A 4.4. táblázat a Maple által megkülönböztetett folyamtípusokat, valamint a megnyitásukra és a lezárásukra szolgáló utasításokat sorolja föl.

Ha megnyitunk egy folyamot, tudatnunk kell a Maple-lel, hogy `READ`, `WRITE` vagy `APPEND` módban kívánjuk használni. A Maple megkülönbözteti a text és a bináris fájlokat: a `TEXT` típus karakterekből, a `BINARY` típus bájtokból álló folyamot jelöl.

Egy egyszerű példa:

```
> fd := fopen( outputfile, WRITE, TEXT );
```

$$fd := 0$$

[1] Ezek az ANSI C nyelv stream fogalmának megfelelői. (*A Fordító megjegyzése.*)

A folyam típusa	Jelentése	Megnyitás/lezárás
`STREAM`	pufferelt fájl	**fopen**, **fclose**
`RAW`	puffereletlen fájl	**open**, **close**
`PIPE`	kétirányú Unix cső	**pipe**, **close**
`PROCESS`	egyik végével egy másik processzhez kapcsolt Unix cső	**popen**, **pclose**
`DIRECT`	az aktuális (`default`) vagy a legfelső szintű (`terminal`) folyam direkt elérése	nem alkalmazható

4.4. táblázat: Maple folyamok megnyitása és lezárása

`fd` a *fájlleíró*, `outputfile` a *fájlnév*, `WRITE` a *fájlmód* és `TEXT` a *fájltípus*. Írjunk be két sort ebbe a fájlba, majd zárjuk le:

```
> writeline( fd, `this is some text` );
                                18

> writeline( fd, `this is some more text` );
                                23

> fclose(fd):
```

A **writeline** eljárás képernyőn megjelenő outputja az utasítás hatására a fájlba beírt karakterek száma. Az `outputfile` fájl tényleg két sorból áll:

```
> ssystem( `cat outputfile` )[2];

                    this is some text
                    this is some more text
```

Így fűzhetünk hozzá egy harmadik sort fájlunkhoz:

```
> fd := fopen( outputfile, APPEND, TEXT ):
> writeline( fd, `third line` );
                                11

> fclose(fd); :
> ssystem( `cat outputfile` )[2];

                    this is some text
                    this is some more text
                    third line
```

Példa kétirányú Unix csőre:

```
> with( process );
```

[*block*, *exec*, *fork*, *kill*, *pclose*, *pipe*, *popen*, *wait*]

Ha Unix csöveket akarunk használni, először be kell töltenünk az őket kezelő segédfüggvényeket tartalmazó process csomagot. Nyissunk meg egy kétirányú csövet:

```
> pds := pipe(); # pipe descriptors
```

$$pds := [0, 1]$$

Két fájlleíróból álló listát kaptunk vissza: az első a csőből való olvasásra, a második a csőbe való írásra használható. A csövek fájltípusa BINARY. Írjunk be valami bináris formátumú szöveget a **writebytes** segítségével:

```
> writebytes( pds[2], `This is a` );
```

$$9$$

```
> convert( ` test`, 'bytes' );
```

$$[32, 116, 101, 115, 116]$$

```
> writebytes( pds[2], " );
```

$$5$$

A cső másik végéből bináris formátumban olvashatunk szöveget, ezt át tudjuk konvertálni ASCII karatersorozattá.

```
> readbytes( pds[1], 14 );
```

$$[84, 104, 105, 115, 32, 105, 115, 32, 97, 32, 116, 101, 115, 116]$$

```
> convert( ", 'bytes' );
```

This is a test

Befejezésül a csövet lezárjuk:

```
> close( pds[1] ):
> close( pds[2] ):
```

Természetesen egy időben több nyitott folyamunk is lehet. Az összes nyitott fájl állapotát az **iostatus**() paranccsal kérdezhetjük le:

```
> restart:
> open( testFile, WRITE );
```

$$0$$

```
> fopen( anotherFile, APPEND, TEXT );
```

$$1$$

```
> process[popen]( rev, WRITE );
```

$$2$$

```
> iostatus();
```

$$[3, 0, 125, [0, testFile, RAW, FD = 11, WRITE, BINARY], [1, anotherFile, STREAM, FP = 135200560, READ, TEXT], [2, rev, PROCESS, FP = 135200640, WRITE, TEXT]]$$

A lista első három eleme: az I/O könyvtár által megnyitott folyamok száma, a pillanatnyilag aktív read utasítások száma és az előző kettő összegének fölső korlátja. A további elemek a nyitott folyamok állapotát írják le.

A Maple támogatja a formázott inputot és outputot is. Ez lehetőséget ad arra, hogy más fölhasználói programok által készített adatokat beolvassunk, vagy a Maple rendszerből írjunk ki adatokat más fölhasználói programok számára. A **printf**, **fprintf**, **sprintf**, **scanf**, **fscanf** és **sscanf** I/O eljárások C nyelvű ekvivalenseikre hasonlítanak. Jelentésüket a 4.5. táblázatban foglaltuk össze.

Parancs	**Jelentés**
fprintf(*stream, format, args*)	az *args* argumentumokat a *stream* folyamba írja ki a *format* formátum szerint
printf(*format, args*)	az *args* argumentumokat a default output folyamba írja ki a *format* formátum szerint
sprintf(*format, args*)	az *args* argumentumokat egy Maple sztringbe írja ki a *format* formátum szerint
fscanf(*stream, format*)	a *stream* folyamból olvas a *format* formátum szerint
scanf(*format*)	a kurrens input folyamból olvas a *format* formátum szerint
sscanf(*string, format*)	a *string* sztringet a *format* formátum szerint dekódolja

4.5. táblázat: Alacsony szintű formázott I/O rutinok

Tehát a

$$\mathbf{printf}(\mathit{format},\ arg_1,\ arg_2,\ \ldots)$$

output eljárás az arg_1, arg_2, ... argumentumokat a default output folyamba írja ki a *format* konverzió-specifikációknak megfelelő formátumban. Például az $\frac{5}{3}$ 15 jegyű numerikus közelítésének kinyomtatását a tizedesvessző utáni 8 tizedesjeggyel és vezető nullákkal feltöltve a következő utasítás valósítja meg:

```
> printf( `%015.8f`, 5/3 );

000001.66666666
```

A formátum-specifikáció általában

`%[flags][width][.precision][code]`

alakú. A konverzió-specifikációk elemei és jelentésük a 4.6. valamint a 4.7. táblázatban láthatók.

A fő különbség a C nyelvű **printf** és a Maple **printf** eljárása között az, hogy a %a konverzió bármilyen sorközök nélküli algebrai kifejezést ki tud nyomtatni az **lprint** utasítással megegyező formátumban, valamint a * karakter nem használható a mezőszélesség meghatározására. Az alábbiakban néhány példát

Flag	Jelentése
-	mezőn belüli balra igazítás
0	a 0 használata vezető karakterként szóköz helyett
+	az előjeles konverzió eredménye mindig előjellel fog kezdődni
szóköz	az előjeles számérték vagy vezető „-" jellel, vagy vezető szóközzel fog kezdődni
#	ha egy számelőjel-flag van jelen, akkor egy alternatív output formátum kerül fölhasználásra

4.6. táblázat: A **printf** flag-jei

mutatunk erre. Az **lprint**() utasítások csak azért szerepelnek, hogy minden egyes prompt jel új sorba kerüljön:

```
> printf( `x=%d y=%4.2f`, 12, 34.3 ); lprint():

x=12      y=34.30

> printf( `%a`, 1/(x^2+1) ); lprint():

1/(x\symbol{94}2+1)

> printf( `%c, %s`, a, abc ); lprint():

a, abc

> printf( `%g, %g`, 1/2, 1/2^16 ); lprint():

.5, 1.525878e-05

> printf( `%f`, 10.^10 ); lprint():

10000000000

> printf( `%a`, 10.^10 ); lprint():

.1000000000e11
```

Ha a Maple karakteres fölhasználói felületével dolgozunk, hasznosak lehetnek az escape karakterek is. Például lehetőséget adnak arra, hogy újsor karaktereket helyezzünk el az outputba, s ezzel elkerüljük az előző példáinkban szereplő **lprint**() utasítás használatát. Az escape karakterek teljes listáját a 4.8. táblázat tartalmazza.

Néhány példa:

```
> printf( `\n%s\n`,
> `string enclosed by newlines\nand a newline inside` );

string enclosed by newlines
and a newline inside
```

Kód	Nyomtatási képe
a	a Maple lprint formátuma
c	egyetlen karakter
d	előjeles decimális szám
e, E	előjeles decimális lebegőpontos szám tudományos jelöléssel
f	előjeles decimális lebegőpontos szám
g, G	előjeles lebegőpontos szám d, e vagy f formátumban, illetve E formátumban, ha G volt specifikálva, annak függvényében, hogy milyen értéket kell nyomtatni
%	egyetlen százalékjel (nincs konverzió)
m	a Maple belső formátuma
o	előjel nélküli oktális szám
p	gépi cím olvasható formában
s	sztring
u	előjel nélküli decimális szám
x, X	előjel nélküli hexadecimális szám, amelyben a, b, c, d, e, f vagy A, B, C, D, E, F jelöli a 9-nél nagyobb számjegyeket

4.7. táblázat: A **printf** konverziós kódjai

```
> printf(`\n%s\n`, `backspacing\b\b\b\b\b\b\bxxxxx`);

backxxxxxing

> printf(`\n%s\n%s\n`, `x\ty\nz`,`w`);

x       y
z
w
```

Az **sscanf**(*string, format*) eljárás az **sprintf** inverze: a megadott *string*-et a *format* formátumnak megfelelően dekódolja. Az egyetlen eltérés a C nyelvű **sscanf**-től az, hogy a Maple változat a dekódolt objektumok listáját adja vissza. Két példa:

```
> sscanf( `x = 123.45, y = 6.7E-8, 9.10`,
> `x = %f, y = %g, %d.%d` );
```

$$[123.45, .67\,10^{-7}, 9, 10]$$

```
> sscanf( `f = x/(1+x^2)`, `f = %a` );
```

$$[\frac{x}{x^2+1}]$$

Általában a formátumspecifikáció a következő alakú:

```
%[*][width][code]
```

Az opcionális * azt jelzi, hogy az objektumot fel kell ismerni, de nem kell visszaadni az eredmény részeként (azaz „eldobható”):

Escape kód	Jelentése
b	backspace
f	lapdobás
n	újsor
r	kocsi vissza
t	horizontális tabulálás
v	vertikális tabulálás
\	backslash
'	egyszeres idézőjel
"	kettős idézőjel
?	kérdőjel

4.8. táblázat: Escape kódok

```
> sscanf(`0-123`,`%*d%d`);
```

$$[-123]$$

```
> sscanf(`0-123`,`%d%d`);
```

$$[0, -123]$$

A `width` mezőszélesség azon karakterek maximális számát jelenti, amelyeket meg kell vizsgálni az aktuális objektum értelmezéséhez. Ez arra is fölhasználható, hogy egy nagyobb egységet két vagy több objektumként ismertessünk föl a függvénnyel:

```
> sscanf( `date = 16121992`, `date = %2d%2d%4d` );
```

$$[16, 12, 1992]$$

A specifikációban használatos karakterek és azok jelentése a 4.9. táblázatban található.

A **parse** parancs egy sztringet Maple utasításként elemez. A `statement` opció határozza meg, hogy a teljes kiértékelést is végre kell-e hajtani:

```
> x := 1:
> parse( `x + 1` );
```

$$x + 1$$

```
> ";
```

$$2$$

```
> parse( `x + 1`, statement );
```

$$2$$

Kód	Jelentése
a	kiértékeletlen Maple kifejezés
c	karakter
d, D	előjeles decimális szám
e, f, g	előjeles decimális lebegőpontos szám
m	belső formátumú Maple kifejezés
n	az eddig olvasott karakterek száma
s	sztring
o, O	előjel nélküli oktális szám
p	egy memóriacímre mutató pointer
u, U	előjel nélküli decimális szám
x, X	előjel nélküli hexadecimális szám, ahol az a,b,c,d,e,f vagy az A,B,C,D,E,F jeleket használjuk a 10, 11, ... számjegyek helyett
%	egyetlen százalékjel
[...]	a kapcsoszárójelek közti karakterhalmazból álló leghosszabb nem üres input sztring
[^...]	a kapcsoszárójelben álló karaktereket *nem* tartalmazó leghosszabb nem üres input sztring

4.9. táblázat: Az **sscanf** eljárás konverziós kódjai

A **filepos**, **feof**, **fflush** és az **fremove** fájlkezelő rutinok az eddig megismert alacsony szintű formázott I/O rutinokkal együtt az adatok be- és kivitelénél „építőkockákként” alkalmazhatók. A lehetőségek megvilágítására néhány példa következik.

Formázott numerikus adatok

Az első példa adott formátumú numerikus adatok exportjáról és importjáról szól. Legyen adott a $(-3,3)$ intervallumba eső lebegőpontos véletlenszámokból álló mátrix. Tegyük föl, hogy ezt soronként szeretnénk kiíratni egy fájlba, mégpedig úgy, hogy a mátrixelemek tabulátor karakterekkel elválasztva 3 jegyű, a tizedespont után 2 jegyet tartalmazó, tudományos jelölésmóddal fölírt számként jelenjenek meg:

```
> data := matrix( 4,3,
> (i,j) -> stats[random,uniform[-3,3]]());
```

$$data := \begin{bmatrix} -.435481985 & -1.073335840 & -.938201558 \\ -.154463138 & .350752314 & 1.480522983 \\ -2.807626668 & 1.337844731 & .625833683 \\ 1.473480224 & -1.441128284 & -1.139547077 \end{bmatrix}$$

```
> data := convert( data, listlist ):
> fd := fopen( datafile, WRITE, TEXT ):
> for row in data do
> for item in row do
> fprintf( fd, `%3.2e\t`, item )
> od:
> fprintf( fd,`\n` ):
> od:
> fclose( fd ):
```

A datafile tartalma ezek után:

```
> ssystem(`cat datafile`)[2];

 -4.35e-01  -1.07e+00  -9.38e-01
 -1.54e-01   3.50e-01   1.48e+00
 -2.80e+00   1.33e+00   6.25e-01
  1.47e+00  -1.44e+00  -1.13e+00
```

Importáljuk ezeket az adatokat kétféle módon listák listájaként: az első esetben a lista álljon az első két oszlopból, a második lista tartalmazza az utolsó oszlopot:

```
> fd := fopen( datafile, READ, TEXT ):
> line := table(): nline := 0:
> while not feof( fd )
> do
> nline := nline + 1:
> line[nline] := fscanf( fd, `%e\t%e\t%*e` )
> od:
> newdata1 := [ seq( line[k], k=1..nline-1 ) ];
```

$$newdata1 := [[-.435, -1.07], [-.154, .350], [-2.80, 1.33], [1.47, -1.44]]$$

Ugyanez mátrix-jelölésmóddal:

```
> setattribute( newdata1, matrix );
```

$$\begin{bmatrix} -.435 & -1.07 \\ -.154 & .350 \\ -2.80 & 1.33 \\ 1.47 & -1.44 \end{bmatrix}$$

Most visszamegyünk a datafile elejére, és beolvassuk a harmadik oszlopot:

```
> filepos( fd, 0 ): # rewind file
> line := table(): nline := 0:
> while not feof( fd )
> do
> nline := nline + 1:
> line[nline] := fscanf(fd,`%*e\t%*e\t%e`)
> od:
> newdata2 := [ seq( line[k], k=1..nline-1 ) ];
```

$$newdata2 := [[-.938], [1.48], [.625], [-1.13]]$$

```
> fclose( fd ):
```

Vegyes numerikus és szöveges adatok

Fájlból való importálásra alternatív módszerként a **readline** parancs is használható. Alkalmazását vegyes (numerikus és szöveges) adatokon is bemutatjuk. Először előállítjuk és egy fájlba exportáljuk a szükséges adatokat:

```
> data := [ seq( [ d.i, stats[random,uniform[0,1]]() ],
> i=1..5 ) ]:
> writedata( datafile, data, [string,float] );
> ssystem(‘cat datafile‘)[2];

d1  .797179
d2  .039169
d3  .088430
d4  .960498
d5  .812920
```

Ezután importáljuk az előző adatokat:

```
> filepos( datafile, 0 ): # ensure starting point
> lines := table(): nline := 0:
> line := readline( datafile ):
> while line <> 0
> do
>   nline := nline + 1:
>   lines[nline] := sscanf( line, ‘%s%f‘ ):
>   line := readline( datafile ):
> od:
> newdata := [ seq( lines[k], k=1..nline ) ];
```

$newdata := [[d1, .797179], [d2, .039169], [d3, .088430], [d4, .960498], [d5, .812920]]$

Különböző fajtájú adatok

Utolsó példánk nem egynemű adatokkal végzett I/O műveleteket mutat be. Adatfájljaink a számok mellett sokszor a fájl tartalmát, létrehozásának dátumát, szerzőjét stb. leíró fejlécet is tartalmaznak. Továbbá az adatok formátuma az egész adathalmazon belül is változhat. Az `agriculture` fájl adatai a Holland Statisztikai Hivatal WWW szerverének mezőgazdasági adatbázisából származnak (URL: http://www.cbs.nl). A fájl tartalma a következő:

```
Title   Livestock in Holland in Recent Years.
Origin  Statistics Netherlands (http://www.cbs.nl)

Livestock        unit    1985    1990    1993    1994
```

```
Cattle      mln    5.2   4.9   4.8   4.7
Pigs        mln    12.4  13.9  15.0  14.6
Sheep       mln    0.8   1.7   1.9   1.8
Chickens    mln    90    93    96    92
Horses      1000   62    70    92    97
Goats       1000   12    61    57    64
```

Most importáljuk és földolgozzuk ezeket az adatokat:

```
> fd := fopen( `agriculture`, READ, TEXT ):
> readline( fd ); # read title
```

Title Livestock in Holland in Recent Years.

```
> title := substring( ", 7..length(") );
```

title := *Livestock in Holland in Recent Years.*

```
> readline( fd ); # read origin
```

Origin Statistics Netherlands (*http* : //*www.cbs.nl*)

```
> origin := substring( ", 8..length(") );
```

origin := *Statistics Netherlands* (*http* : //*www.cbs.nl*)

```
> readline( fd ):  # read empty line
> fscanf( fd, `%s%s%d%d%d%d` );
```

[*Livestock*, *unit*, 1985, 1990, 1993, 1994]

```
> header := ":
> readline( fd ):  # read empty line
> livestock := table():
> while not feof( fd )
> do
> line := fscanf(fd,`%s%a%f%f%f%f`):
> if line <> 0 then
> line := subs( mln=10^6, line ):
> fctr := line[2]:
> livestock[line[1]] := map( z -> fctr*z, line[3..6] )
> fi
> od:
> livestock[Horses], livestock[Sheep]; # two examples
```

[62000., 70000., 92000., 97000.],
[800000.0, $.17000000\,10^7$, $.19000000\,10^7$, $.18000000\,10^7$]

4.6. Kódgenerálás

Harmadfokú polinomok gyökeit a Maple **solve** eljárásával találhatjuk meg.

```
> polyeqn := x^3 - a*x=1;
```

$$polyeqn := x^3 - a\,x = 1$$

```
> sols:= solve(polyeqn, x );
```

$$sols := \frac{1}{6}\%1^{1/3} + 2\frac{a}{\%1^{1/3}},$$
$$-\frac{1}{12}\%1^{1/3} - \frac{a}{\%1^{1/3}} + \frac{1}{2}I\sqrt{3}\left(\frac{1}{6}\%1^{1/3} - 2\frac{a}{\%1^{1/3}}\right),$$
$$-\frac{1}{12}\%1^{1/3} - \frac{a}{\%1^{1/3}} - \frac{1}{2}I\sqrt{3}\left(\frac{1}{6}\%1^{1/3} - 2\frac{a}{\%1^{1/3}}\right)$$
$$\%1 := 108 + 12\sqrt{-12a^3 + 81}$$

Ha a Maple által kiszámolt hasonló analitikus eredményeket egy FORTRAN programban akarjuk föhasználni, a **fortran** eljárással a Maple-lel lefordíttathatjuk őket. Egyszerre az összes megoldásra ez nem lehetséges, mivel a FORTRAN-ban nem létezik a kifejezés-sorozat adattípus. Ezért vagy minden megoldást külön kell lefordítani, vagy előre el kell helyezni a megoldásokat egy vektorban vagy listában.

```
> sol1 := sols[1]; # take 1st solution
```

$$sol1 := \frac{1}{6}(108 + 12\sqrt{-12a^3 + 81})^{1/3} + 2\frac{a}{(108 + 12\sqrt{-12a^3 + 81})^{1/3}}$$

Ha a **fortran** eljárás hívásakor az `optimize` kulcsszót is megadjuk, akkor (a lehetőségek szerint) optimalizált FORTRAN kódot kapunk. Ha duplapontosságú FORTRAN kódot kívánunk, a `precision` és a `mode` opciókat állítsuk be a `double` értékre. A `precision` opció a lebegőpontos számok, a `mode` opció a matematikai függvények fordítását vezérli:

```
> fortran( sol1, 'optimized',
> precision=double, mode=double );

      t1 = a**2
      t6 = (108.D0+12.D0*dsqrt(-12.D0*t1*a+81.D0))**(1.D0/3.D0)
      t9 = t6/6.D0+2.D0*a/t6
```

A **cost** eljárással vizsgálható meg, hogy mennyire volt sikeres a kódoptimalizáció:

```
> readlib( cost )( sol1 );
```

$$5\,additions + 15\,multiplications + 8\,functions$$

```
> cost( optimize(sol1) );
```

$$3\,additions + 8\,multiplications + divisions + 4\,functions$$
$$+\,4\,assignments$$

Figyeljük meg, hogy a Maple csak a belső reprezentáció szerinti fölírásban keresi a közös tényezőket és hatványokat. Például az alábbi kifejezésre triviálisan alkalmazható

$$(a+b+c)(a+b) \quad \longrightarrow \quad t \overset{\text{def}}{=} b+c; \quad (a+t)\,t$$

egyszerűsítést nem találja meg a Maple.

```
> optimize( (a+b+c)*(b+c) );
```

$$t3 = (a+b+c)(b+c)$$

A **C** Maple eljárás C kódot generál. Ha az outputot a *file.c* fájlba szeretnénk átirányítani, kiegészítésképpen használjuk a `filename` = *file.c* argumentumot. Az előzőekben tárgyalt **fortran** függvénynél is megvan ez a lehetőség:

```
> readlib( C ): # load the C routine
> C( sol1, filename = 'file.c' ): # generate code
```

Egy fájlban több formulát is tárolhatunk:

```
> C( sol1, 'optimized', filename = 'file.c' ):
```

Az alábbi listázásból látható, hogy a rendszer az első megoldás optimalizált kódját hozzáfűzte a fájl végéhez (az első sort három részre tördeltük, hogy kiférjen az oldalon).

```
t0 = pow(108.0+12.0*sqrt(-12.0*a*a*a+
     81.0),1.0/3.0)/6+2.0*a/pow(108.0+
     12.0*sqrt(-12.0*a*a*a+81.0),1.0/3.0);
t1 = a*a;
t4 = sqrt(-12.0*t1*a+81.0);
t6 = pow(108.0+12.0*t4,1.0/3.0);
```

A **fortran** és a **C** rutinok alkalmazhatók névvel bíró tömbökre, egyenletek listáira és komplett Maple eljárásokra is. A következő példák a kódgenerálásnál használható további opciókat is demonstrálják.

```
> fortran( [ x=Pi*ln(t), y=x^2-sqrt(gamma) ],
> precision=single, mode=complex );

      x = 0.3141593E1*clog(t)
      y = x**2-csqrt(0.5772157E0)

> fortran( [ p=x, q=x^2, r=p+q ], 'optimized' );

      p = x
      q = x**2
      r = x+q
```

Az utóbbi értékadások megkaphatók a következő Maple eljárás FORTRAN nyelvű megfelelőjének segítségével is:

```
> f := proc(x) global p,q,r:
> p := x:  q := x^2:  r := p+q end:
> fortran( f, 'optimized' );

      real function f(x)
      real x

      common/global/p,q,r
      real p
      real q
      real r
```

```
      p = x
      q = x**2
      r = x+q
    return
    end
```

A változók kezelésénél még ennél is rugalmasabb a rendszer. Például előírhatjuk, hogy a p legyen lokális változó:

```
> fortran( [ p=x, q=x^2, r=p+q ], 'optimized',
> locals=[p] );

    q = x**2
    r = x+q
```

A most következő, kicsit látványosabb példában a Maple eljárásból előállított FORTRAN kód a tömbök kezelését, az output nyomtatását és a hibakezelést illusztrálja:

```
> trigvalues := proc(x)
>   local i,trigs:  global Digits:
>   if not type( x, numeric ) then
>     ERROR( `non-numeric input` )
>   elif Digits < 15 then
>     Digits := 15
>   fi:
>   trigs := array( 1..4, sparse,
>     [ (1)=sin(x), (2)=cos(x), (3)=tan(x) ] ):
>   for i to 4 do print( trigs[i] ) od:
>   trigs
> end:
> fortran( trigvalues, precision=single );

    subroutine trigvalues(x,crea\_par)
    real x
    real crea\_par(4)

    integer i
    real trigs(4)

    common/global/Digits
    integer Digits

      if ( .not. type(x,numeric)) then
        write(6,*) 'non-numeric input'
        call exit(0)
      else
        if (Digits .lt. 15) then
          Digits = 15
        endif
      endif
      trigs(1) = sin(x)
      trigs(2) = cos(x)
      trigs(3) = tan(x)
      trigs(4) = 0
```

```
      do 1000 i = 1,4,1
        write(6,*) trigs(i)
1000  continue
      crea\_par(1) = trigs(1)
      crea\_par(2) = trigs(2)
      crea\_par(3) = trigs(3)
      crea\_par(4) = trigs(4)
      return
    return
    end
```

A 4.10. és a 4.11. táblázat a **fortran**, illetve a **C** eljárás opcióit sorolja föl.

Opció	**Jelentése**	**Default**
`digits` = n	a konstansokat n jegyre kerekítse	7, 16 decimális jegy
`filename` = *file*	az outputot a *file* fájlhoz fűzze hozzá	terminál
`globals` = g	g legyen globális változó	minden változó globális
`locals` = l	l legyen globális változó	semmi sem lokális
`mode` = *type*	a függvényeket *type* típusúra konvertálja (`single`, `double`, `generic` vagy `complex`)	`single`
`optimize`	optimalizált kód	nincs optimalizálás
`parameters` = p	a számítási folyamat p paraméterének megadása	értékadás baloldalán nem szereplő változók
`precision` = p	a p pontosság megadása (`single`,`double`)	`single`

4.10. táblázat: A **fortran** eljárás opciói

A Maple a **latex** eljárást kínálja LaTeX kód generálására. Ha ugyanabba a fájlba egymás után több formulára akarunk LaTeX kódot generálni, akkor használjuk az **appendto** eljárást. A következő példában az előző egyenlet két megoldásának kódját állítjuk elő. Azt is megmutatjuk, hogyan lehet a **latex** eljárást hozzáigazítani az igényeinkhez. A LaTeX matematikai módjába való átmenettel egészítjük ki a kódgenerálást. Föltételezzük, hogy a karakteres alapú Maple fölhasználói interfészt használjuk.

```
> #ensure that all remember tables are present
> readlib( latex ):
> `latex/special_names`['beginmath'] :=
>    `\\begin{displaymath}`:
> `latex/special_names`['endmath'] :=
>    `\\end{displaymath}`:
> # suppress messages, prompts, etc.
```

```
> interface( quiet=true ):
> appendto( `file.tex` ):
> latex(beginmath); latex(sols[1]); latex(endmath);
> writeto(terminal); interface( quiet=false );
> quit
```

Opció	Jelentése	Default
`ansi`	ANSI C kompatibilis kód	a Kernighan–Ritchie-féle `cc` fordító
`digits` = *n*	a konstansokat *n* jegyre kerekítse	7, 16 decimális jegy
`filename` = *file*	az outputot a *file* fájlhoz fűzze hozzá	terminál
`globals` = *g*	*g* legyen globális változó	minden változó globális
`locals` = *l*	*l* legyen globális változó	semmi sem lokális
`optimize`	optimalizált kód	nincs optimalizálás
`parameters` = *p*	a számítási folyamat *p* paraméterének megadása	értékadás baloldalán nem szereplő változók
`precision` = *p*	a *p* pontosság megadása (`single`,`double`)	`single`

4.11. táblázat: A **C** eljárás opciói

Amennyiben a `screenwidth` interface változó értéke az alapértelmezés szerinti 80, akkor a `file.tex` így néz ki:

```
\begin{displaymath}
1/6\,\sqrt[3]{108+12\,\sqrt{-12\,{a}^{3}+81}}+2\,{\frac{a}{\sqrt[3
]{108+12\,\sqrt {-12\,{a}^{3}+81}}}}
\end{displaymath}
```

A LaTeX által kiszedett formula a következő:

$$1/6\,\sqrt[3]{108+12\,\sqrt{-12\,{a}^{3}+81}}+2\,{\frac{a}{\sqrt[3]{108+12\,\sqrt {-12\,{a}^{3}+81}}}}$$

Ha kényelmetlennek találjuk a `latex(beginmath)` és a `latex(endmath)` utasítások hozzáadását minden egyes alkalommal, akkor írhatunk egy kis programot:

```
mylatex := proc(x)
global `latex/special_names`:
readlib(latex):
`latex/special_names`['beginmath'] :=
`\\begin{displaymath}`:
```

```
`latex/special_names`['endmath'] :=
`\\end{displaymath}`:
latex(beginmath); latex(x); latex(endmath);
end;
```

és hívhatjuk a **mylatex** eljárást a rendes **latex** helyett. A

```
> macro( latex=mylatex ):
```

utasítás végrehajtása után a **mylatex** helyett használható a **latex** név is. Ha a formulák túl nagyok ahhoz, hogy elférjenek egy sorban, akkor a kapott LaTeX fájlt magunknak kell átszerkeszteni és a formulákat sorokra tördelni. A munkalapos interfész esetében ez automatikusan megtörténik.

Teljes Maple munkalapok is elmenthetők LaTeX formátumban. A programmal adott (és az `ftp.maplesoft.com`-ról ftp-vel is letölthető) stílusfájlok használatával így szépen kiszedett dokumentumokat kapunk. Nagyobb képletek esetén ez még kényelmesebb, mivel lehetővé teszi a sorszélesség interaktív beállítását, és gondoskodik a formulák ennek megfelelő részekre tördeléséről.

4.7. A Maple hozzáigazítása igényeinkhez

Szöveg alapú fölhasználói interfész esetén a Maple hosszabb számítások közben üzenetet küld a fölhasznált memóriáról és a számítási időről. Ezek az üzenetek arról informálnak bennünket, hogy a Maple szekció kezdetétől fogva mennyi számítási időt és memóriát használtunk. Az üzenetek formátuma a következő:

$$bytes\ used = <x>, \quad alloc = <y>, \quad time = <z>$$

Itt $<x>$ egyenlő a Maple szekció kezdete óta fölhasznált byte-okban meghatározott összes memóriával, $<y>$ a pillanatnyilag allokált memória mérete, $<z>$ pedig a Maple szekció kezdetétől eltelt, másodpercekben kifejezett számítási időt mondja meg. Nézzünk egy numerikus integrálás közben kiadott üzenetsorozatot:

```
> evalf( Int( ln(x)*ln(1-x), x=0..1 ) );

bytes used=1010508, alloc=851812, time=0.41
bytes used=2010880, alloc=1572576, time=1.05
bytes used=3011416, alloc=1834672, time=1.75
bytes used=4011692, alloc=1900196, time=2.47
```

$$.3550659332$$

Karakteres fölhasználói felület esetén a **kernelopts** eljárással szabályozhatjuk a *bytes used* üzenetek gyakoriságát.

```
> oldprintbytes := kernelopts( printbytes = false );
```

$$oldprintbytes := true$$

Ezután nem kapunk *bytes used* üzeneteket. A munkalapos interfész esetében a megfelelő menüpontok kijelölésével dönthetjük el, hogy a munkalap alján legyen-e ilyen üzenet-régió. Az előző paranccsal egyszerre mentettük el és állítottuk át a `printbytes` kernelopció értékét. Ezért a régi érték könnyen visszaállítható.

```
> kernelopts( printbytes = oldprintbytes ):
```

Bár egy Maple szekción belül a fölhasznált komputerszavak száma folyamatosan nő, ez mégsem jelenti azt, hogy az aktuálisan használt memória mérete is ilyen gyorsan növekszik. A rendszer időnként szemétgyűjtést („garbage collection") végez, hogy a szükségtelen objektumok eltakarítása után felszabaduló memóriát újra föl tudja használni. Minden szemétgyűjtéskor megjelenik a képernyőn a *bytes used* üzenet (ha csak le nem tiltottuk). A szemétgyűjtést mi is kérhetjük a **gc** eljárás segítségével:

```
> gc():
```

A `gcfreq` kernelopcióval vezérelhetjük a szemétgyűjtések gyakoriságát. Ha ezt írjuk:

```
> oldfreq := kernelopts( gcfreq = 10^6 );
```

$$oldfreq := 2500000$$

akkor minden újabb egy millió szó fölhasználása után történik szemétgyűjtés. Látható, hogy a default érték 250000 szó (1 szó 4 vagy 8 bájtot jelent, az adott gépen ezt a **kernelopts(wordsize)** paranccsal deríthetjük ki).

Ha a szöveges interfész esetén zavaróak ezek az üzenetek, a rendszert két módon is elhallgattathatjuk:

```
> kernelopts( printbytes = false ):
```

Ez a korábban megismert módszer. A `bytesused`, `bytesalloc` és a `cputime` kernelopciókkal a rendszer memória- és időfölhasználása bármikor lekérdezhető (a további kernelopciók megismeréséhez írjuk be a **?kernelopts** parancsot).

```
> kernelopts( bytesused , bytesalloc, cputime );
```

$$64595100, 5504016, 37.469$$

A másik drasztikusabb módszer a Maple elcsendesítésére:

```
> interface( quiet = true ):
```

Ennek mellékhatásaként a Maple prompt is eltűnik, és ha az eredményeket a **writeto**-val fájlba írjuk, az input sorok sem kerülnek be a fájlba. A Maple számára a hallgatás abszolut hallgatást jelent!

Ugyanezzel az **interface** eljárással az output megjelenítését is befolyásolhatjuk. Kívánságunk szerint változtathatjuk a prompt jelet, letilthatjuk, hogy közös részkifejezéseket a Maple automatikusan cimkézzen, vagy balra igazított egydimenziós output-formátumot írhatunk elő. Néhány példa:

```
> interface( prompt = `--->` ); # an arrow as a prompt
```

```
---> solve( x^3+x+1, x); #labels for common subexpressions
```

$$-\frac{1}{6}\%2+\frac{2}{(108+12\sqrt{93})^{1/3}},$$
$$\frac{1}{12}\%2-\frac{1}{(108+12\sqrt{93})^{1/3}}+\frac{1}{2}I\sqrt{3}\,(-\frac{1}{6}\%2-\frac{2}{(108+12\sqrt{93})^{1/3}}),$$
$$\frac{1}{12}\%2-\frac{1}{(108+12\sqrt{93})^{1/3}}-\frac{1}{2}I\sqrt{3}\,(-\frac{1}{6}\%2-\frac{2}{(108+12\sqrt{93})^{1/3}})$$
$$\%1:=\frac{1}{(108+12\sqrt{93})^{1/3}}$$
$$\%2:=(108+12\sqrt{93})^{1/3}$$

```
---> interface( labeling = false ); # no labels anymore

---> solve(x^3+x+1,x);
```

$$-\frac{1}{6}(108+12\sqrt{93})^{1/3}+\frac{2}{(108+12\sqrt{93})^{1/3}},\ \frac{1}{12}(108+12\sqrt{93})^{1/3}$$
$$-\frac{1}{(108+12\sqrt{93})^{1/3}}$$
$$+\frac{1}{2}I\sqrt{3}\,(-\frac{1}{6}(108+12\sqrt{93})^{1/3}-\frac{2}{(108+12\sqrt{93})^{1/3}}),$$
$$\frac{1}{12}(108+12\sqrt{93})^{1/3}-\frac{1}{(108+12\sqrt{93})^{1/3}}$$
$$-\frac{1}{2}I\sqrt{3}\,(-\frac{1}{6}(108+12\sqrt{93})^{1/3}-\frac{2}{(108+12\sqrt{93})^{1/3}})$$

```
---> Example := (1-x)/(1+x):

---> Example; # 2-dimensional layout
```

$$\frac{1-x}{1+x}$$

```
--->  interface( prettyprint = 0 );
---> Example; # 1-dimensional layout
(1-x)/(1+x)
---> # reset prettyprint to high resolution
```

```
---> interface( prettyprint = 2 ):

---> interface( verboseproc = 2 ): # make Maple verbose

---> readlib( unassign ); # procedure body is printed

proc()
local i, j, n;
global assign;
options `Copyright (c) 1990 by the University of Waterloo.
All rights reserved.`;
    i := false;
    for n in args do
        if n = 'assign' then i := true
        elif type(n, indexed) then `unassign/indexed`(n)
        else assign(n,n)
        fi
    od;
    if i then assign := evaln(assign) fi;
    NULL
end

---> interface( verboseproc = 3 ):

---> # make Maple more verbose

---> readlib( ln );

proc(x)
options `Copyright (c) 1992 by the University of Waterloo.
All rights reserved.`;
if nargs <> 1 then ERROR(`expecting 1 argument, got `.nargs)
 elif type(x, 'complex(float)') then evalf('ln'(x))
 elif x = 0 then ERROR(`singularity encountered`)
 elif type(x, 'function') and op(0, x) = exp and Im(op(1,x)) = 0
      then  op(1, x)
 elif type(x, 'rational') and numer(x) = 1 and 0 < x
      then  -ln(denom(x))
 elif type(x, 'constant^numeric') and signum(op(1, x)) = 1
      then op(2, x)*ln(op(1, x))
 elif type(x, '`^`') and (type(op(2, x), 'integer')
  or is(op(2, x), 'integer')) and (signum(0, Re(op(1, x)),1) = 1
  or member(signum(0, Im(op(1, x)), 0), {-1, 1}))
      then op(2, x)*ln(op(1, x))
else ln(x) := 'ln'(x)
fi
end
# (-1) = (-1)^(1/2)*Pi
# (1) = 0
# (infinity) = infinity
# ((-1)^(1/2)) = 1/2*(-1)^(1/2)*Pi
# (-(-1)^(1/2)) = - 1/2*(-1)^(1/2)*Pi
```

Az utolsó két példa azt mutatja, hogy a könyvtári függvények kódja emlékezőtábláikkal együtt megtekinthető. Csak a kernel függvényeinek C kódja nem hozzáférhető.

Egy kézhezálló interfész változó az `errorbreak`. Azt írja elő, hogy mi történjék, ha a rendszer egy hibás input sort olvas be. Lehetséges értékeit a 4.12. táblázatban soroltuk föl:

Érték	**Eredménye**
0	a hiba jelzése után a beolvasás folytatódik
1	szintaktikus hiba után a beolvasás megszakad
2	bármilyen hiba hatására megszakad a beolvasás

4.12. táblázat: Az `errorbreak` értékei

A **?interface** parancs megadja az összes interfész változót, lehetséges értékeikkel és fölhasználási területeikkel együtt.

4.8. Gyakorlatok

1. Állapítsa meg az Olvasó, hogy számítógépén hol található a Maple könyvtár, és olvassa be az `isolate.m` fájlt a könyvtárból két módon:

   (a) a **read** paranccsal

   (b) a **readlib** fölhasználásával, miután a `verboseproc` interfész változót kettőre állította.

2. Vizsgáljuk meg az `echo` interfész változót.

   - Először is hozzuk létre a `readfile` nevű fájlt a home könyvtárunkban. A fájl tartalma legyen egyetlen Maple parancs:

```
b := 2;
```

   - Másodszor hozzuk létre a `testfile` fájlt a home könyvtárunkban. A fájl tartalma legyen a következő utasítássorozat:

```
interface( echo = X );
a := 1;
read readfile;
quit
```

   - Harmadszorra, ha az általunk használt platform megengedi, akkor indítsuk el a Maple-t a home könyvtárunkból a következő paranccsal:

```
maple < testfile
```

úgy, hogy az X értékét a testfile-ban rendre $0, 1, 2, 3$ és 4-re írja át.

- Végül olvassuk be a `testfile`-t egy Maple szekcióba a **read** eljárás segítségével, miközben X értékét a testfile első sorában $0, 1, 2, 3$, 4-re változtatjuk.

3. Hozzunk létre egy fájlt a következő tartalommal:

```
I := 1; # szintaktikusan helyes, de futásidőben hibaüzenet
x := 2; # helyes inputsor
wrong name := 3 # szintaktikai hiba
y := 4; # helyes inputsor
```

Az `errorbreak` interface változó minden lehetséges értékére derítsük ki, hogy mi történik, ha a fájlt beolvassuk egy Maple szekcióba.

4. Tekintsük a következő, három sort tartalmazó adatfájlt:

```
1    2
3    4
5    6
```

Olvassuk be a fájlt, konvertáljuk mátrixszá, transzponáljuk a mátrixot, és nyomtassuk ki az adatokat a következő formában:

```
1    2    3
4    5    6
```

5. Számítsuk ki az x^{x^x} kifejezés második deriváltját, és fordítsuk le a kapott eredményt FORTRAN-ra. Vizsgáljuk meg a kódoptimalizálás hatását.

5.

Polinomok és racionális függvények

A Maple kedvenc matematikai struktúrái a polinomok és a racionális függvények. Ez a fejezet bevezeti az Olvasót az egy- és többváltozós polinomok, a racionális függvények használatába, valamint a különböző formák közötti konverziókba. A kulcsszavak a következők: legnagyobb közös osztó (gcd), faktorizálás, kifejtés, többváltozós polinomok lexikografikus, illetve fokszám szerinti rendezése, racionális függvények normalizálása és parciális törtekre bontása.

5.1. Egyváltozós polinomok

A polinomok és a racionális függvények (vagyis a polinomok hányadosai) a Maple rendszer kedvenc adattípusai. A számítógépes algebrai rendszer ezeket nagy sebességgel és teljes általánosságban kezeli. Az egyszerűség kedvéért kezdetben egyváltozós polinomokat tekintünk (a változót x-szel jelöljük). Matematikai szempontból ezek ekvivalensek az

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

alakú szimbolikus kifejezésekkel. Ezt a polinom *kifejtett kanonikus alakjának*, az $a_0, a_1, \ldots a_{n-1}, a_n$ mennyiségeket a polinom *együtthatóinak* nevezzük. Ha $a_n \neq 0$, akkor a_n-t *főegyütthatónak*, az n természetes számot pedig a polinom *fokszámának* nevezzük. Később látni fogjuk, hogy a polinomok együtthatóira nem kell túl sok kikötést tennünk, de pillanatnyilag arra az esetre szorítkozunk,

amikor ezek „közönséges" (például egész, racionális, valós vagy komplex) számok. Azt mondjuk, hogy az x egy polinomja *gyűjtött alakú*, ha az x változó összes megegyező kitevőjű hatványának együtthatói a polinomban összegyűjtve (csak egyszer) szerepelnek. A kifejtett kanonikus alaktól mindez pusztán abban különbözik, hogy az x különböző hatványai nem föltétlenül fokszám szerint csökkenő sorrendben következnek. Egy példa:

```
> restart;
> p1 := -3*x + 7*x^2 - 3*x^3 + 7*x^4; # collected form
```

$$p1 := -3\,x + 7\,x^2 - 3\,x^3 + 7\,x^4$$

```
> type( p1, polynom );
```

$$true$$

```
> leading_coefficient = lcoeff(p1);
```

$$leading_coefficient = 7$$

```
> degree = degree(p1);
```

$$degree = 4$$

```
> p2 := 5*x^5 + 3*x^3 + x^2 - 2*x + 1;
```

$$p2 := 5\,x^5 + 3\,x^3 + x^2 - 2\,x + 1$$

```
> 2*p1 - 3*p2 + 3;
```

$$11\,x^2 - 15\,x^3 + 14\,x^4 - 15\,x^5$$

```
> p1 * p2;
```

$$(-3\,x + 7\,x^2 - 3\,x^3 + 7\,x^4)\,(5\,x^5 + 3\,x^3 + x^2 - 2\,x + 1)$$

```
> expand(");
```

$$-17\,x^6 + 11\,x^4 - 20\,x^3 + 13\,x^2 - 3\,x + 56\,x^7 + 4\,x^5 - 15\,x^8 + 35\,x^9$$

A számok szorzásával ellentétben a polinomok összeszorzását a Maple nem hajtja végre automatikusan. Erre külön utasítanunk kell az **expand** eljárással. Ez hátrányosnak tűnik, de valójában nem az. A $(3x+5)^{10}$ faktorizált formát jobb változatlanul hagyni a számítás során, amíg csak lehetséges. Ez sokkal világosabb, mint

$$59049\,x^{10} + 984150\,x^9 + 7381125\,x^8 + \cdots + 58593750\,x + 9765625,$$

a polinom kifejtett formája. Bár a Maple két polinom szorzatát helyesen számolja ki, a válaszban szereplő tagok sorrendje általában se nem növekvő, se nem csökkenő. Mindez gyakran vezet nehezen kiolvasható és értelmezhető kifejezésekhez. A polinomok tagjait a rendszer hatékonysági megfontolásokból (számítási idő, memória fölhasználás) nem rendezi valamilyen előre definiált standard sorrendbe. Ha a tagokat valóban az x hatványai szerint csökkenő sorrendben akarjuk látni, akkor a **sort** parancsot kell használnunk.

```
> sort(");
```

$$35\,x^9 - 15\,x^8 + 56\,x^7 - 17\,x^6 + 4\,x^5 + 11\,x^4 - 20\,x^3 + 13\,x^2 - 3\,x$$

A **sort** megváltoztatja a polinomnak megfelelő belső adatstruktúrát. Ez szükséges is, hiszen a Maple bármely kifejezésből vagy részkifejezésből csak egy példányt tárol a memóriában, pontosabban az *egyszerűsítési táblájában.* Minden egyes új kifejezésre a rendszer „belső algebrája" ellenőrzi, hogy a kifejezés elemi műveletekkel egyszerűsíthető-e egy olyan kifejezéssé, amely már korábban előfordult, és így megtalálható az egyszerűsítési táblában. Ha ez az eset áll fönn, akkor a Maple figyelmen kívül hagyja az új kifejezést, és az egyszerűsítési táblában találhatót használja helyette. Az ellenkező esetben a kifejezést összes részkifejezéseivel együtt elhelyezi az egyszerűsítési táblában. Összeg tagjainak, illetve szorzat tényezőinek megváltozása példa olyan elemi egyszerűsítésekre, melyeket a kifejezések matematikai ekvivalenciájának vizsgálatakor használ a rendszer.

```
> restart;
> p := 1+x+x^3+x^2; # random order
```

$$p := 1 + x + x^3 + x^2$$

```
> x^3+x^2+x+1; # no rearrangement, so still random order
```

$$1 + x + x^3 + x^2$$

```
> q := (x-1)*(x^3+x^2+x+1); # the same for subexpressions
```

$$q := (x - 1)(1 + x + x^3 + x^2)$$

```
> sort( p ); # rearrange w.r.t. descending degree order
```

$$x^3 + x^2 + x + 1$$

```
> q; # subexpression of q has also changed
```

$$(x - 1)(x^3 + x^2 + x + 1)$$

A Maple számos eljárást kínál a polinomok kezelésére. Néhányat közülük megvizsgálunk. Amíg mást nem mondunk, *p*1 és *p*2 jelentse a korábban bevezetett két polinomot.

```
> 'p1' = p1, 'p2' = p2;
```

$$p1 = -3x + 7x^2 - 3x^3 + 7x^4,\ p2 = 5x^5 + 3x^3 + x^2 - 2x + 1$$

A **coeff** eljárással a polinomot alkotó valamely egytagnak vagy x valamely hatványának együtthatóját határozhatjuk meg:

```
> coeff( p2, x^3 ), coeff( p2, x, 2 );
```

$$3, 1$$

A főegyüttható az **lcoeff**, az x^0 együtthatója a **tcoeff** segítségével kapható meg:

```
> lcoeff( p2, x ), tcoeff( p2, x );
```

$$5, 1$$

Az együtthatók és a nekik megfelelő hatványok sorozatát a **coeffs** eljárás szolgáltatja:

```
> coeffs( p2, x, 'powers' );
```

$$-2, 1, 3, 1, 5$$

```
> powers; # assigned in the previous command
```

$$x, 1, x^3, x^2, x^5$$

A Maple az **lcoeff**, **tcoeff**, **coeffs** és a **degree** eljárások használatánál föltételezi, hogy a polinomok gyűjtött alakúak.

```
> lcoeff( x^2 - x*(x-1), x );
```

$$0$$

```
> coeffs( (x^2-1) * (x^2+1), x );

Error, invalid arguments to coeffs
> coeffs( collect((x^2-1)*(x^2+1),x), x );
```

$$-1, 1$$

```
> degree( x^2 - x*(x-1), x );
```

$$2$$

A polinomokkal kapcsolatos egyik legalapvetőbb művelet a *maradékos osztás*. A Maple-nek ennek végrehajtására két eljárása is van, a **quo** és a **rem** (**quo**tient = hányados, **rem**ainder = maradék).

```
> q := quo( p2, p1, x, 'r' );
```

$$q := \frac{5}{7}x + \frac{15}{49}$$

A **quo** hívásánál a negyedik argumentum megadása opcionális, amennyiben szerepel, a polinomosztás maradékát kapja értékül.

```
> r;
```

$$1 - \frac{53}{49}x^3 + x^2 - \frac{53}{49}x$$

```
> testeq( p2 = expand(q*p1+r) ); # test equality
```

$$\mathit{true}$$

```
> rem( p2, p1, x, 'q' );
```

$$1 - \frac{53}{49}x^3 + x^2 - \frac{53}{49}x$$

```
> q; # quotient has been set in previous command
```

$$\frac{5}{7}x + \frac{15}{49}$$

Ne feledkezzünk meg az aposztrófok használatáról ezen eljárások negyedik argumentuma körül, hogy ezzel elnyomjuk az argumentum nem kívánt teljes kiértékelését.

Egy másik fontos függvény a **gcd**, amely a racionális számtest fölötti két polinom legnagyobb közös osztóját határozza meg:

```
> gcd( p1, p2 );
```

$$x^2 + 1$$

A legnagyobb közös osztó meghatározása olyannyira fontos művelet a különböző matematikai számítások során, hogy a számítógépes algebrai rendszerek tervezői komoly erőfeszítéseket tettek hatékony GCD algoritmusok tervezésére és implementálására (v. ö. [37, 76, 117]).

Még bonyolultabb algoritmus a polinomok szorzattá alakítása (faktorizációja). Az érdeklődő Olvasónak [112, 114, 115] tanulmányozását javasoljuk. A **factor** eljárás a racionális együtthatós polinomokat ($\mathbb{Q}$ fölött) irreducibilis polinomok szorzataként írja föl:

```
> polynomial := expand( p1*p2 );
```

$$polynomial := -17\,x^6 + 11\,x^4 - 20\,x^3 + 13\,x^2 - 3\,x + 56\,x^7 + 4\,x^5 - 15\,x^8 + 35\,x^9$$

```
> factor( polynomial );
```

$$x\,(7\,x - 3)\,(5\,x^3 - 2\,x + 1)\,(x^2 + 1)^2$$

```
> factor( polynomial, I );
```

$$(5\,x^3 - 2\,x + 1)\,(7\,x - 3)\,x\,(x + I)^2\,(x - I)^2$$

Az egész és a racionális számoktól különböző tartományokban is végezhetünk műveleteket polinomokkal (például véges testek, az algebrai számok vagy függvénytestek fölött). Az előző polinomok faktorizációja ezen tartományok fölött a következő:

- Szorzattá alakítás a Gauss egészek $\mathbb{Z}[i]$ gyűrűje fölött:

```
> factor( polynomial, I );
```

$$(5\,x^3 - 2\,x + 1)\,(7\,x - 3)\,x\,(x + I)^2\,(x - I)^2$$

- Szorzattá alakítás a 0 és 1 elemekből álló kételemű $\mathbb{Z}_2$ test fölött:

```
> sort( polynomial mod 2 ); # polynomial over {0,1}
```

$$x^9 + x^8 + x^6 + x^4 + x^2 + x$$

```
> Factor( polynomial ) mod 2;
```

$$(x + 1)^6\,(x^2 + x + 1)\,x$$

```
> expand(") mod 2;
```

$$x^9 + x^8 + x^6 + x^4 + x^2 + x$$

Figyeljük meg a **Factor** függvényben használt nagybetűt. Használatának célja az, hogy elnyomja az egészek fölötti szorzattá alakítást. A **Factor**(...) pusztán arra szolgál, hogy reprezentálja a polinom szorzattá alakítását, és valójában a **mod** operátor az, ami a tényleges számítást elvégzi. Különben először megkapnánk a polinom faktorizált formáját a racionális számok fölött, és a rendszer erre alkalmazná a modulo aritmetikát:

```
> factor( polynomial) mod 2;
```

$$x\,(x + 1)\,(x^3 + 1)\,(x^2 + 1)^2$$

- Szorzattá alakítás a $\mathbb{GF}(4)$ Galois test fölött, pontosabban a $\mathbb{Z}_2$ testnek az x^2+x+1 irreducibilis polinommal való algebrai bővítése fölött:

```
> alias( alpha = RootOf( z^2 + z + 1, z ) ):
> Factor( polynomial, alpha ) mod 2;
```

$$(x+\alpha)\,(x+1)^6\,(x+\alpha+1)\,x$$

A helykijelölő mechanizmus alkalmazása más polinomokon működő eljárások, így a **Gcd**, **Divide**, **Expand** stb. esetében is szokásos. Nézzünk néhány példát:

```
> Gcd(p1,p2) mod 2;
```

$$x^3+x^2+x+1$$

```
> Expand( p1 * p2 ) mod 2;
```

$$x^9+x^8+x^6+x^4+x^2+x$$

```
> q := Quo( p2, p1, x, 'r' ) mod 2;
```

$$q := x+1$$

```
> r;
```

$$x^3+x^2+x+1$$

```
> Expand( p2 - q*p1 - r ) mod 2;
```

$$0$$

5.2. Többváltozós polinomok

Az egynél több változóval rendelkező polinomokat többváltozós polinomoknak nevezzük. A következő kétváltozós polinomban az x és y változókat használtuk.

```
> polynomial := 6*x*y^5 + 12*y^4 + 14*y^3*x^3
>    - 15*x^2*y^3 + 9*x^3*y^2 - 30*x*y^2 - 35*x^4*y
>    + 18*y*x^2 + 21*x^5;
```

$$\mathit{polynomial} := 6\,x\,y^5+12\,y^4+14\,y^3\,x^3-15\,x^2\,y^3+9\,x^3\,y^2-30\,x\,y^2$$
$$-35\,x^4\,y+18\,y\,x^2+21\,x^5$$

Láthatjuk, hogy a Maple-nek nincs különösebb elképzelése a polinom tagjainak, illetve az egyes tagokban előforduló ismeretleneknek a sorrendjéről. Elfogadja az általunk beírt sorrendet. Ha valami más sorrendet akarunk, ismét a **sort** eljárást alkalmazhatjuk:

```
> sort( polynomial, [x,y], 'plex' );
```

$$21\,x^5-35\,x^4\,y+14\,x^3\,y^3+9\,x^3\,y^2-15\,x^2\,y^3+18\,x^2\,y+6\,x\,y^5-30\,x\,y^2+12\,y^4$$

Itt a tagok *valódi lexikografikus* (pure lexicographic) rendezését kaptuk, amelyet az

$$x^i y^j \prec x^{i'} y^{j'} \iff i < i' \text{ or } (i = i' \text{ and } j < j')$$

föltételek határoznak meg. Eszerint

$$1 \prec y \prec y^2 \prec \ldots \prec x \prec x\,y \prec x^2 \ldots.$$

Használhatjuk a teljes fokszám szerinti (total degree) rendezést is, amelyet a következő módon definiálunk:

$$x^i y^j \prec x^{i'} y^{j'} \iff i + j < i' + j' \text{ or } (i + j = i' + j' \text{ and } i < i')$$

Ekkor

$$1 \prec y \prec x \prec y^2 \prec x\,y \prec x^2 \prec y^3 \prec x\,y^2 \prec x^2\,y \prec x^3 \ldots.$$

Ez az elrendezés a **sort** eljárás alapértelmezése.

```
> sort( polynomial );
```

$$14\,x^3\,y^3 + 6\,x\,y^5 + 21\,x^5 - 35\,x^4\,y + 9\,x^3\,y^2 - 15\,x^2\,y^3 + 12\,y^4 + 18\,x^2\,y \\ - 30\,x\,y^2$$

A fenti eredményt tekinthetjük az x határozatlan olyan polinomjának, amelynek együtthatói y polinomjai. A polinomot a **collect** eljárással hozhatjuk ennek megfelelő alakra:

```
> collect( polynomial, x );
```

$$21\,x^5 - 35\,x^4\,y + (14\,y^3 + 9\,y^2)\,x^3 + (18\,y - 15\,y^3)\,x^2 + (-30\,y^2 + 6\,y^5)\,x \\ + 12\,y^4$$

Másrészt a `polinomial` kifejezés fölfogható az y változó polinomjának is:

```
> collect( polynomial, y );
```

$$6\,x\,y^5 + 12\,y^4 + (-15\,x^2 + 14\,x^3)\,y^3 + (9\,x^3 - 30\,x)\,y^2 \\ + (-35\,x^4 + 18\,x^2)\,y + 21\,x^5$$

Sok egyváltozós polinomot kezelő Maple eljárás ugyanolyan jól működik a többváltozós esetben is. Csak néhány példát mutatunk, azt ajánljuk, hogy az Olvasó maga is kísérletezzen tovább a többváltozós polinomokkal.

```
> coeff( polynomial, x^3 ), coeff( polynomial, x, 3 );
```

$$14\,y^3 + 9\,y^2,\ 14\,y^3 + 9\,y^2$$

```
> coeffs( polynomial, x, 'powers' ); powers;
```

$$-30\,y^2 + 6\,y^5,\ 12\,y^4,\ 18\,y - 15\,y^3,\ 14\,y^3 + 9\,y^2,\ -35\,y,\ 21$$

$$x,\ 1,\ x^2,\ x^3,\ x^4,\ x^5$$

```
> coeff( coeff( polynomial, x^3 ), y^2 );
```

$$9$$

```
> settime := time(): # start timing
> factor( polynomial );
```

$$(3x^2 - 5xy + 2y^3)(7x^3 + 6y + 3xy^2)$$

```
> Factor( polynomial ) mod 7;
```

$$2y(3y^3 + 3xy + x^2)(2 + xy)$$

```
> cpu_time := (time()-settime) * seconds;
```

$$cpu_time := .670\, seconds$$

Nincs ember, aki meg tudja közelíteni ezt a számítási sebességet. Kérje csak meg az Olvasó matematikus barátait és kollegáit, hogy próbálják végrehajtani a fenti szorzattá alakítást papírral és ceruzával! Komoly matematikai tudásanyag és bonyolult programozási munka áll a **factor** és a **Factor** eljárások mögött.

5.3. Racionális függvények

Az x ismeretlen racionális függvénye egy olyan kifejezés, amelyet f/g alakban írhatunk föl, ahol f és g x polinomja, és g nem egyenlő 0-val. Egy példa:

```
> f := x^2 + 3*x + 2: g := x^2 + 5*x + 6: f/g;
```

$$\frac{x^2 + 3x + 2}{x^2 + 5x + 6}$$

A racionális kifejezés számlálóját és nevezőjét a **numer** és a **denom** eljárásokkal határozhatjuk meg (**numer**ator = számláló, **denom**inator = nevező).

```
> numer("), denom(");
```

$$x^2 + 3x + 2,\ x^2 + 5x + 6$$

Figyeljük meg, hogy a racionális számokkal végzett számításokkal ellentétben a Maple nem egyszerűsíti a racionális kifejezéseket oly módon, hogy a számláló és a nevező relatív prím legyen. Az automatikus egyszerűsítéseket csak akkor hajtja végre a rendszer, ha közvetlenül fölismer egy közös tényezőt:

```
> ff := (x-1)*f; gg := (x-1)^2*g;
```

$$ff := (x - 1)(x^2 + 3x + 2)$$

$$gg := (x - 1)^2 (x^2 + 5x + 6)$$

```
> ff/gg;
```

$$\frac{x^2 + 3x + 2}{(x - 1)(x^2 + 5x + 6)}$$

Ha a racionális kifejezést ilyen formára akarjuk hozni, akkor a **normal** eljárást kell alkalmaznunk. Hatékonysági megfontolásokból az eljárás a számlálóban és a nevezőben talált tényezőket igyekszik változatlanul megőrizni, amennyiben ez lehetséges.

```
> normal( f/g );
```

$$\frac{x + 1}{x + 3}$$

```
> normal( ff/gg );
```

$$\frac{x+1}{(x+3)\,(x-1)}$$

Három oka van annak, hogy a racionális kifejezések normalizálását nem végzi el a rendszer automatikusan:

▷ A normalizálás nem mindig ad egyszerű eredményt. Az $(x^{10000}-1)/(x-1)$ például sokkal tömörebb, mint normalizált formája, a tízezer tagú polinom.

▷ Nagyon időigényes lenne a számolás közben előforduló minden egyes racionális kifejezésre elvégezni a normalizációt.

▷ Elképzelhető, hogy a fölhasználó más műveleteket, például parciális törtekre bontást akar végezni.

A fejezetet záró példa azt mutatja, hogyan egyszerűsíthetünk a többváltozós racionális kifejezés számlálójában és nevezőjében előforduló közös tényezővel. (Ellenőrizzük az eredményt a részkifejezések szorzattá alakításával!)

```
> f := 161*y^3 + 333*x*y^2 + 184*y^2 + 162*x^2*y
> + 144*x*y + 77*y + 99*x + 88:
> g := 49*y^2 + 28*x^2*y + 63*x*y + 147*y + 36*x^3
> + 32*x^2 + 117*x + 104:
> ratexpr := f/g;
```

$$ratexpr := (161\,y^3 + 333\,x\,y^2 + 184\,y^2 + 162\,x^2\,y + 144\,x\,y + 77\,y + 99\,x + 88)/ (49\,y^2 + 28\,x^2\,y + 63\,x\,y + 147\,y + 36\,x^3 + 32\,x^2 + 117\,x + 104)$$

```
> normal( ratexpr );
```

$$\frac{18\,x\,y + 23\,y^2 + 11}{4\,x^2 + 7\,y + 13}$$

5.4. Konverziók

A polinomok *Horner-elrendezése* főleg numerikus számolások esetén célszerű, ugyanis az aritmetikai műveletek száma ebben az alakban minimális. Ezt a korábban definiált *p*1 polinommal illusztráljuk. Az elvégzett összeadások és szorzások számát a **cost** eljárással határozhatjuk meg:

```
> p1;
```

$$-3\,x + 7\,x^2 - 3\,x^3 + 7\,x^4$$

```
> readlib(cost)( p1 );
```

$$3\ additions + 10\ multiplications$$

```
> convert( p1, 'horner' );
```

$$(-3+(7+(-3+7x)x)x)x$$

```
> cost(");
```

$$3\,additions + 4\,multiplications$$

A racionális kifejezések lánctörtté alakítása szintén meggyorsíthatja a számolást (**con**tinued **frac**tion = lánctört).

```
> q := (x^3 + x^2 - x + 1) / p1;
```

$$q := \frac{x^3+x^2-x+1}{-3x+7x^2-3x^3+7x^4}$$

```
> cost( q );
```

$$6\,additions + 13\,multiplications + divisions$$

```
> convert( q, 'confrac', x );
```

$$\frac{1}{7}\frac{1}{x-\frac{10}{7}+\frac{24}{7}\frac{1}{x+\frac{11}{6}+\frac{1}{9}\frac{1}{x-\frac{71}{24}+\frac{429}{64}\frac{1}{x+\frac{17}{8}}}}}$$

```
> cost(");
```

$$7\,additions + 4\,divisions$$

A **hornerform** és a **confracform** eljárások, melyek a Horner-elrendezést, illetve a lánctörtté alakítást végzik, a numerikus közelítéseket számító `numapprox` csomagban találhatók meg. Erről részletesebben lásd a 11.2. alfejezetet.

A racionális kifejezések parciális törtekre való bontását (**convert** to **par**tial **frac**tion) is könnyedén elvégezhetjük a Maple-lel:

```
> convert( q , 'parfrac', x );
```

$$\frac{143}{87}\frac{1}{-3+7x}-\frac{1}{3}\frac{1}{x}+\frac{1}{29}\frac{3+7x}{x^2+1}$$

A parciális törtekre bontott alakból majdnem közvetlenül kiolvasható a racionális kifejezés határozatlan integrálja:

```
> integrate( q, x );
```

$$-\frac{1}{3}\ln(x)+\frac{143}{609}\ln(-3+7x)+\frac{7}{58}\ln(x^2+1)+\frac{3}{29}\arctan(x)$$

Figyeljük meg, hogy az előző példában a Maple föltételezi, hogy a racionális számtestben dolgozunk. Ha lebegőpontos vagy komplex parciális törtekre bontást kívánunk, negyedik argumentumként adjuk meg a `real`, illetve a `complex` kulcsszót.

```
> convert( q , 'parfrac', x, real );
```

$$-\frac{.3333333334}{x}+\frac{.2348111659}{x-.4285714286}$$
$$+.1428571429\frac{.7241379320+1.689655174x}{x^2+1.}$$

```
> convert( q , 'parfrac', x, complex );
```

$$-\frac{.3333333334}{x}+\frac{.1206896552+.05172413794\,I}{x+I}$$
$$+\frac{.1206896552-.05172413794\,I}{x-1.\,I}$$
$$+\frac{.2348111659-.1733520995\,10^{-10}\,I}{x-.4285714286}$$

Ha a nevező összes elsőfokú tényezőjét meg akarjuk kapni, az alaptest algebrai lezárása fölött is végeztethetjük a fölbontást. Az algoritmus részletes leírása [23]-ban található. Ez a teljes parciális törtekre bontás (**convert** to **full partial fraction**) a következő módon végezhető el:

```
> convert( q , 'fullparfrac', x );
```

$$-\frac{1}{3}\frac{1}{x}+\left(\sum_{_\alpha=\%1}\frac{\frac{595}{2523}-\frac{3}{58}_\alpha+\frac{581}{5046}_\alpha^2}{x-_\alpha}\right)$$

$$\%1 := \mathrm{RootOf}(-3+7_Z-3_Z^2+7_Z^3)$$

Az itt szereplő harmadfokú polinom zérushelyeit gyökös kifejezésekként fölírva az összegzés elvégezhető.

```
> convert( ", radical );
```

$$-\frac{1}{3}\frac{1}{x}+\frac{143}{609}\frac{1}{x-\frac{3}{7}}+\frac{\frac{7}{58}-\frac{3}{58}I}{x-I}+\frac{\frac{7}{58}+\frac{3}{58}I}{x+I}$$

Természetesen szabad segítenünk is a Maple-nek, ha sejtjük vagy tudjuk, hogy milyen testbővítéssel célszerű próbálkozni. Ezt egyszerűen beírhatjuk negyedik argumentumként:

```
> 1 / ( x^4 - 5*x^2 + 6):
```

$$\frac{1}{x^4-5x^2+6}$$

```
> convert( ", 'parfrac', x, sqrt(2) );
```

$$\frac{1}{x^2-3}+\frac{1}{4}\frac{\sqrt{2}}{x+\sqrt{2}}+\frac{1}{4}\frac{\sqrt{2}}{-x+\sqrt{2}}$$

```
> convert( "", 'parfrac', x, {sqrt(2),sqrt(3)} );
```

$$\frac{1}{6}\frac{\sqrt{3}}{x-\sqrt{3}}-\frac{1}{6}\frac{\sqrt{3}}{x+\sqrt{3}}+\frac{1}{4}\frac{\sqrt{2}}{x+\sqrt{2}}+\frac{1}{4}\frac{\sqrt{2}}{-x+\sqrt{2}}$$

Utolsó példaként egy többváltozós racionális kifejezés parciális törtekre bontását határozzuk meg:

```
> ratfun := (x-a)/(x^5+b*x^4-c*x^2-b*c*x);
```

$$ratfun := \frac{x-a}{x^5+b\,x^4-c\,x^2-b\,c\,x}$$

```
> convert( ratfun, 'parfrac', x );
```

$$\frac{a}{b\,c\,x}-\frac{-b^2\,c-b\,c\,a+b\,c\,x+c\,x\,a-c\,x^2+x^2\,b^2\,a}{c\,(c+b^3)\,(x^3-c)}-\frac{b+a}{b\,(c+b^3)\,(x+b)}$$

```
> map( collect, ", x );
```

$$\frac{a}{b\,c\,x}-\frac{(-c+b^2\,a)\,x^2+(b\,c+c\,a)\,x-b^2\,c-b\,c\,a}{c\,(c+b^3)\,(x^3-c)}-\frac{b+a}{b\,(c+b^3)\,(x+b)}$$

```
> convert( ratfun, 'parfrac', x, true );
```

$$\frac{a}{b\,c\,x}-\frac{-b\,c-c\,a+a\,x^2\,b+a\,x^3}{b\,c\,(x^4+b\,x^3-c\,x-b\,c)}$$

Az utolsó parancsban opcionális negyedik argumentumként a `true`-t adtuk meg, ezzel jelezve, hogy nem szükséges `ratfun`-ra a **normal** eljárást alkalmazni, és hogy a nevező már a kívánt faktorizált alakú.

5.5. Gyakorlatok

1. Tekintsük az $\frac{x^4+x^3-4\,x^2-4\,x}{x^4+x^3-x^2-x}$ racionális kifejezést. Alakítsuk át ezt a kifejezést a következő alakok mindegyikébe:

(a) $\frac{(x+2)(x+1)(x-2)}{x^3+x^2-x-1}$

(b) $\frac{x^4+x^3-4\,x^2-4\,x}{x(x-1)(x+1)^2}$

(c) $\frac{(x+2)(x-2)}{(x-1)(x+1)}$

(d) $\frac{x^2}{(x-1)(x+1)}-4\,\frac{1}{(x-1)(x+1)}$

2. Tekintsük az előző feladatban használt racionális kifejezést.

(a) Adjuk meg a kifejezést lánctört alakban.

(b) Számítsuk ki a parciális törtekre bontását.

3. Legyen f az $x^7+x^5+2\,x^3+2\,x^2+3\,x+2$ polinom.

(a) Alakítsuk f-et szorzattá $\mathbb{Z}_5$ és $\mathbb{Z}_7$ fölött.

(b) Miért következik az előző szorzattá alakításból, hogy f irreducibilis az egészek gyűrűje fölött? (Figyeljük meg a tényezők fokszámát!) Ellenőrizzük a Maple segítségével, hogy f valóban irreducibilis $\mathbb{Z}$ fölött.

4. Néhány prímtest fölötti faktorizáció:

   (a) Alakítsuk szorzattá $(x^2 - 2)$-t $\mathbb{Z}_2$ fölött.

   (b) Alakítsuk szorzattá $(x^3 - 3)$-at $\mathbb{Z}_3$ fölött.

   (c) Alakítsuk szorzattá $(x^5 - 5)$-öt $\mathbb{Z}_5$ fölött.

   (d) Alakítsuk szorzattá $(x^{23} - 23)$-at $\mathbb{Z}_{23}$ fölött.

   (e) Mostanra kialakulhatott az az érzésünk, hogy $(x^p - p)$ minden p prímre szorzattá alakítható. Ellenőrizzük ezt a sejtést az előzőektől különböző prímszámokkal. Meg tudjuk magyarázni az eredményt?

5. Legyen $f = 2x^4 - 3x^2 + x + 4$ és $g = 2x^5 - 6x^4 - 4x^3 + x^2 - 3x - 2$. Tekintsük ezeket $\mathbb{Z}_7$ fölötti polinomoknak, és számoljuk ki f és g legnagyobb közös osztóját. Határozzuk meg az $sf + tg = \gcd(f, g)$ egyenlőséget kielégítő $\mathbb{Z}_7$ fölötti s és t polinomokat is (a legnagyobb közös osztó természetesen $\mathbb{Z}_7$ fölött értendő).

6. Ha megkérjük a Maple-t, hogy alakítsa szorzattá az $x^{2458}+x^{1229}+1$ polinomot $\mathbb{Z}$ fölött, akkor a rendszer várhatóan panaszkodni fog, hogy nincs elég memóriája, vagy több órás számolás után még mindig nem találja meg az eredményt. Vizsgáljuk meg közelebbről a problémát, és nézzük meg, hogyan segíthetnénk a rendszernek. Például az **isprime** eljárással kideríthetjük, hogy 1229 prímszám, vagyis polinomunk $x^{2p} + x^p + 1$ alakú, ahol p prímszám. A Maple segítségével határozzuk meg az ilyen polinomok faktorizációját kis prímekre, és fogalmazzunk meg valamilyen sejtést a szorzattá alakított polinom általános alakjáról. Gyakorlott matematikusok tudják, hogy az $x^n - 1$ polinom, ahol n természetes szám, $\mathbb{Z}$ fölött a $\phi_k(x)$ *ciklotomikus polinomok* szorzatává alakítható az alábbi módon:

$$x^n - 1 = \prod_{k|n} \phi_k(x)$$

A ciklotomikus polinomok irreducibilisek $\mathbb{Z}$ fölött. Ennek fölhasználásával nem lehet túl nehéz korábbi sejtésünk bizonyítása.

6.

Belső adatábrázolás és helyettesítés

Ebben a fejezetben részletesen leírjuk a polinomok és racionális függvények belső ábrázolását. Mindennek nem csak elméleti jelentősége van. Fontos ugyanis ismerni az adatok belső ábrázolását, ha meg akarjuk érteni a különböző adatstruktúrák közötti konverziókat és a helyettesítést.

Általánosítjuk a racionális kifejezések fogalmát, és megmagyarázzuk, hogy a polinomokra és racionális kifejezésekre definiált eljárásokat hogyan terjeszthetjük ki általánosabb matematikai struktúrákra.

Foglalkozunk a típusellenőrzéseket és a formulák részeinek kiválasztását végző Maple eljárásokkal is. Ezek a rutinok segítenek a különböző adatstruktúrák vizsgálatában vagy a velük végzett műveletekben.

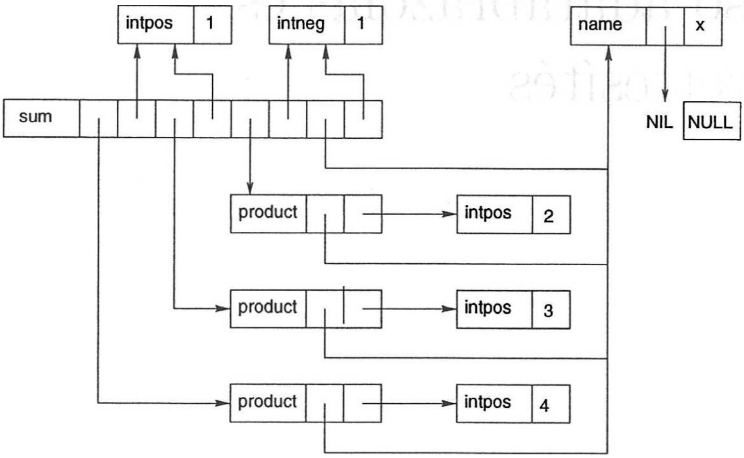
Végül megvizsgáljuk a szekvenciális és az egyidejű helyettesítést, valamint azt, hogy ezek hogyan használhatók kifejezések egyszerűsítésére.

6.1. Polinomok belső ábrázolása

Az előző fejezetben már láttunk olyan Maple eljárásokat, amelyek polinomokat alakítanak át egyik alakjukból a másikba. Annak érdekében, hogy teljesen megérthessük ezeket, valamint az ezután ismertetendőket, közelebbről is megvizsgáljuk a polinomok belső ábrázolását a Maple-ben.

Az $x^4+x^3-x^2-x$ polinomot vesszük példaként. Kifejtett kanonikus alakjának belső reprezentációja a 6.1. ábrán látható adatvektorokból áll, ezek gráfelméleti

terminológiát használva *irányított aciklikus gráfot* (DAG = directed acyclic graph) alkotnak. A 6.2. ábrán látható adatvektort úgy értelmezhetjük, mintha egy-egy



6.1. ábra: $x^4 + x^3 - x^2 - x$ belső reprezentációja

kifejezést és annak numerikus együtthatóját tartalmazó párokból állna. Az adatvektor a következő összeget reprezentálja:

$$\text{coeff}_1 \times \text{expr}_1 + \text{coeff}_2 \times \text{expr}_2 + \cdots .$$

sum	expr1	coeff1	expr2	coeff2	....

6.2. ábra: Összeg belső reprezentációja

A 6.3. ábra vektora hasonló fölépítésű. Ez az

$$\text{expr}_1^{\text{expon1}} \times \text{expr}_2^{\text{expon2}} \times \cdots$$

szorzatot írja le.

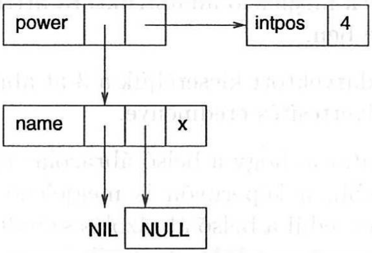
product	expr1	expon1	expr2	expon2	....

6.3. ábra: Szorzat belső reprezentációja

A 6.1. ábrán látható NIL pointer azt jelzi, x szabad változó. Az üres kifejezéssorozatot jelölő NULL Maple kifejezésre mutató pointer azt jelzi, hogy x-hez nem rendeltünk semmilyen attributumokat.

Az első szinten a belső ábrázolás azt mutatja, hogy a Maple négy tag összegének tekinti az $x^4 + x^3 - x^2 - x$ polinomot, a tagok x^4, x^3, x^2 és x rendre az $1, 1, -1$ és -1 együtthatókkal. Ezen komponensek belső ábrázolása adatvektorokkal történik. Meglepőnek tűnhet, hogy az x^4 tag belső ábrázolásában egy

olyan adatvektort használ a rendszer, melynek fejlécében `product` áll, ezenkívül egy pointer mutat az x változóra, egy másik pedig a 4 kitevőre. Ehelyett talán a 6.4. ábrán látható adatvektort várt volna az Olvasó.



6.4. ábra: x^4 hibás belső reprezentációja

Ez a fajta belső ábrázolás is létezik a Maple-ben, de ha a kitevő numerikus konstans, akkor a Maple egyszerűsítő eljárása átalakítja az előzőekben látott `product` struktúrává. A rendszer az $x^4+x^3-x^2-x$ polinom részkifejezéseiként csak az $x^4, x^3, x^2, -x^2, x$ és a $-x$ polinomokat ismeri fel. Ami a Maple-t illeti, az x^4+x^3 vagy éppenséggel az x^4-x nem részkifejezése az $x^4+x^3-x^2-x$ polinomnak. Az a tény, hogy a Maple nem ismeri föl az összes matematikai értelemben vett részkifejezést, rendkívül fontos szerepet játszik a helyettesítés során. Egyedül ezzel magyarázhatók az olyan helyettesítések, mint például

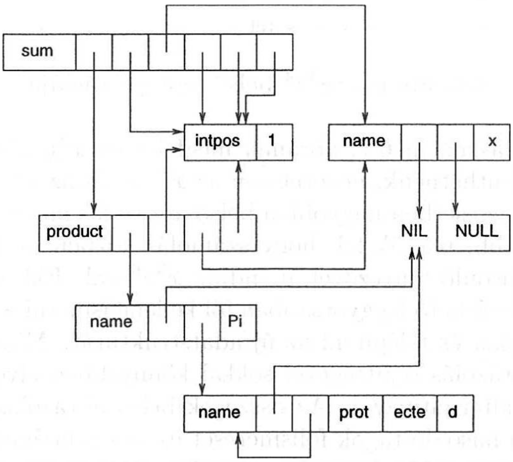
```
> subs( 1 = 7, x^4 + x^3 - x^2 - x );
```

$$7x^4+7x^3-x^2-x$$

```
> subs( 1 = 3, Pi*x + x + 1 );
```

$$3\pi^3x^3+3x+9$$

Vessünk egy pillantást az utolsó példára. A $\pi x + x + 1$ belső ábrázolása a 6.5. ábrán látható DAG.



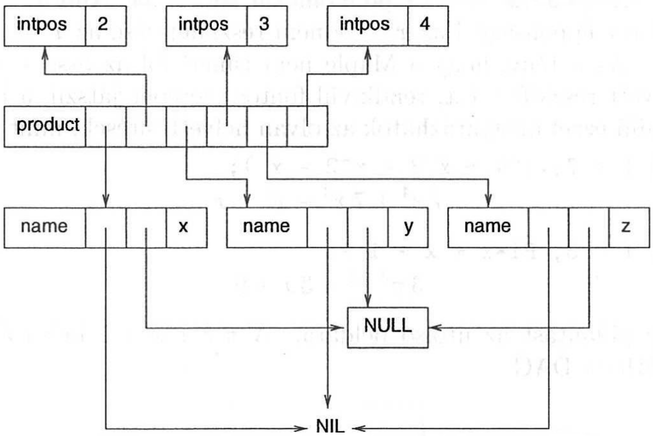
6.5. ábra: $\pi x + x + 1$ belső reprezentációja

Ez az irányított aciklikus gráf azt is tükrözi, hogy `Pi` a `protected` attributummal bír. Ez az attributum egy érvényes Maple kifejezés, ti. egy név. A nevek karakterei 32 bites gépeken legfeljebb négy karakterből álló csoportokban tárolódnak. A `protected`-nek megfelelő adatstruktúra attributuma saját neve, így ez is védett név a Maple-ben.

Ha az 1-et ábrázoló adatvektort kicseréljük a 3-at ábrázoló adatvektorra, akkor érthetővé válik a helyettesítés eredménye.

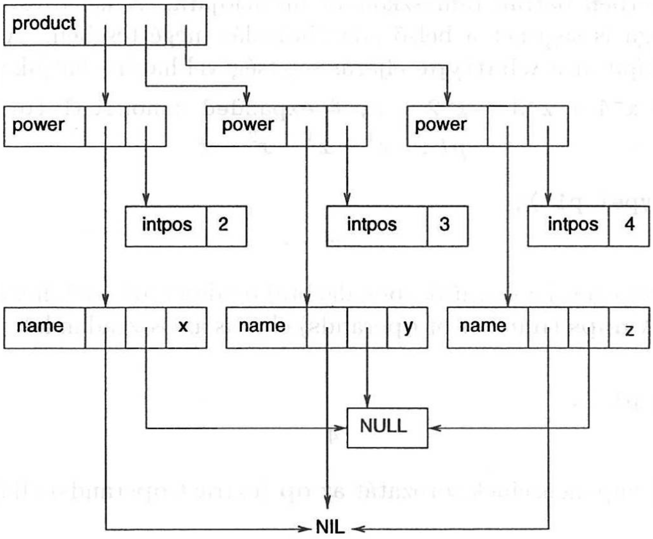
A két utolsó példa mutatja, hogy a belső ábrázolás jelentősen eltérhet a számunkra sokkal ismerősebb, a képernyőn is megjelenő külső ábrázolástól. A Maple számára viszont egyedül a belső ábrázolás számít. Két kifejezés akkor és csak akkor azonos, ha ugyanaz a DAG tartozik hozzájuk. Az egyszerűsítés a rendszer számára nem más, mint a megfelelelő DAG-ok tarnszformálása.

Mielőtt folytatnánk, gondoljuk meg, hogy a Maple tervezői vajon miért választhatták az összegekre és a hatványokra az itt tárgyalt belső ábrázolást. Tekintsük az $x^2y^3z^4$ kifejezés belső ábrázolását (6.6. ábra).



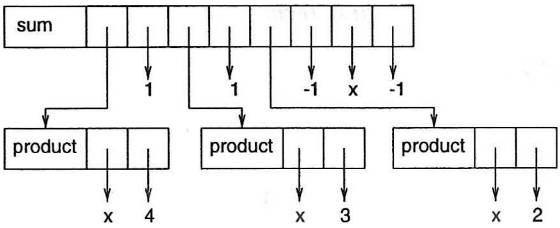
6.6. ábra: $x^2y^3z^4$ belső reprezentációja

Alternatív megoldásként a 6.7. ábrának megfelelően $x^2y^3z^4$-t három hatvány szorzatának is tekinthetnénk, nevezetesen az x^2, y^3 és az x^4 szorzatának. Látható, hogy ebben az esetben nagyobb a belső ábrázolás memóriaigénye. De ami ennél még fontosabb, tegyük fel, hogy számolás közben az $x^2y^3z^4$-t meg kell szorozni valami hasonló tényezővel, mondjuk x^5z^6-nal. Ekkor a Maple egyszerűsítő eljárásának a lehető leggyorsabban föl kellene ismerni a közös tényezőket, összeadni a kitevőket és fölépíteni az új adatstruktúrát. Mindez a Maple által használt belső ábrázolás segítségével sokkal könnyebben elvégezhető, mint az általunk javasolt alternatívával. Az összegek belső ábrázolása könnyen végrehajthatóvá teszi a hasonló tagok fölismerését és az együtthatók összeadását is. Röviden összefoglalva, a Maple rendszer belső adatábrázolását arra optimalizálták, hogy a polinomokkal való számolás minél gyorsabb legyen.



6.7. ábra: $x^2y^3z^4$ hibás belső reprezentációja

Felejtsük el egy pillanatra, hogy a Maple minden (rész)kifejezést csak egyszer tárol. Ezesetben a belső ábrázolásnak megfelelő irányított aciklikus gráf egy *fa* lenne. A példánknak megfelelő *reprezentációs fa* a 6.8. ábrán látható.



6.8. ábra: $x^4 + x^3 - x^2 - x$ reprezentációs fája

Elhagytuk az 1, -1 és x építőelemeknek megfelelő adatvektorokat, ezeket a formula elemi részeinek tekintjük. Az $x^4+x^3-x^2-x$ reprezentációs fája szintén jól mutatja, hogy a polinom alkotórészei $x^4, x^3, -x^2$ és $-x$, továbbá az x^2 részkifejezés. A reprezentációs fa legfontosabb előnye, hogy sokkal áttekinthetőbb, nem annyira kusza, mint az irányított aciklikus gráf. Lényeges különbség, hogy míg az irányított aciklikus gráf egy csúcspontot használ az azonos részkifejezések ábrázolására, addig a reprezentációs fában ezek külön részfákként szerepelnek. A fentieket szem előtt tartva a reprezentációs fák a Maple kifejezések belső struktúrájának adekvát, könnyen megérthető leírását adják.

Az Olvasó feje már bizonyára zúg a sok adatvektortól, irányított aciklikus gráftól és reprezentációs fától. Talán azon mereng, hogy képes lesz-e valaha is megérteni a Maple kifejezések belső ábrázolását. Vigasztalásul annyi, hogy

a legtöbb esetben bátran támaszkodhat intuíciójára. A kétséges esetekben a rendszer maga is segíthet a belső adatábrázolás megértésében. A kifejezések felszíni adattípusát a **whattype** eljárás segítségével határozhatjuk meg:

```
> p1 := x^4 + x^3 - x^2 - x; # expanded canonical form
```

$$p1 := x^4 + x^3 - x^2 - x$$

```
> whattype( p1 );
```

$$+$$

A „+" jel segítségével a számítógépes algebrai rendszer azt jelzi, hogy a kifejezés egy összeg. A **nops** (**n**umber of **op**erand**s**) eljárás az összeadandók számát adja meg:

```
> nops( p1 );
```

$$4$$

A kifejezés komponenseinek sorozatát az **op** (extract **op**erands) eljárás szolgáltatja:

```
> op(p1);
```

$$x^4,\ x^3,\ -x^2,\ -x$$

Megkaphatjuk a részkifejezéseket egyenként is. Például az első tag az x negyedik hatványa:

```
> `first term` := op(1,p1);
```

$$\textit{first term} := x^4$$

```
> whattype(");
```

$$\hat{}$$

```
> op("");
```

$$x,\ 4$$

Itt a „^" jel a hatvány adattípusnak felel meg.

A harmadik tag, $-x^2$, a -1 és az x^2 hatvány szorzata. A Maple rendszer a szorzat adattipust „*"-gal jelzi.

```
> `third term` := op(3,p1);
```

$$\textit{third term} := -x^2$$

```
> whattype(");
```

$$*$$

```
> op( `third term` );
```

$$-1,\ x^2$$

```
> op( [3,2], p1 ); # 2nd operand of 3rd operand of p1
```

$$x^2$$

Nemcsak a polinomok, hanem akármilyen Maple kifejezések is szétszedhetők hasonló módon. De ne feledjük, hogy a Maple az azonos részkifejezéseket csak egyszer tárolja. A fenti példában az x karakter négy különböző helyen fordul elő, de a belső ábrázolásban ezeknek egyetlen Maple objektum felel meg. Ha az $x^4 + x^3 - x^2 - x$ polinomban x-et y-nal helyettesítjük, akkor az $y^4 + y^3 - y^2 - y$ polinomot kapjuk.

Ha az Olvasó jobban meg szeretné ismerni a Maple-t, és a reprezentációs fák helyett az irányított aciklikus gráfokat szeretné tanulmányozni, a **dismantle** eljárás segítségével ismerheti meg a részleteket. Ez a Maple kifejezések szerkezetét részkifejezéseikkel és azok hosszával együtt jeleníti meg:

```
> readlib( dismantle ): # load library function
> dismantle( x^4+x^3-x^2-x );

SUM(9)
   PROD(3)
      NAME(4): x
      INTPOS(2): 4
   INTPOS(2): 1
   PROD(3)
      NAME(4): x
      INTPOS(2): 3
   INTPOS(2): 1
   PROD(3)
      NAME(4): x
      INTPOS(2): 2
   INTNEG(2): -1
   NAME(4): x
   INTNEG(2): -1
```

6.2. Általánosított racionális kifejezések

A következő Maple parancsok a racionális függvények belső ábrázolását mutatják meg:

```
> r := (y^2-1) / (y-1);
```

$$r := \frac{y^2 - 1}{y - 1}$$

```
> type( r, ratpoly ); # check if rational expression
```

$$true$$

```
> whattype( r );
```

$$*$$

```
> op(r);
```

$$y^2 - 1, \frac{1}{y - 1}$$

```
> op(2,r);
```

$$\frac{1}{y-1}$$

```
> whattype(");
```

$$\wedge$$

```
> op("");
```

$$y-1,\ -1$$

```
> normal( r ); # normal form
```

$$y+1$$

Ismét csak láthatjuk, hogy a belső adatábrázolás eltérhet a munkalapon vagy a terminál képernyőjén látható külső formátumtól. A racionális kifejezés a számlálónak és a nevező (-1)-dik hatványának szorzata. Ha a kifejezést az $\mathbb{R}(y)$ test elemének tekintjük, akkor normalizált alakja az $y+1$ polinom. Valós függvényekként tekintve azonban az $\frac{y^2-1}{y-1}$ és az $y+1$ különbözőek.

Vegyük a következő kifejezést:

```
> r := (sin(x)^2-1)/(sin(x)-1);
```

$$r := \frac{\sin(x)^2-1}{\sin(x)-1}$$

```
> type( r, ratpoly );
```

false

Ez a Maple szerint sem racionális függvény. Ha azonban $\sin x$-et y-nal helyettesítjük, akkor visszakapjuk az előző racionális függvényt. Erre rájöhetünk úgy is, ha a $\frac{\sin^2 x-1}{\sin x-1}$ belső adatábrázolását vizsgáljuk az **op**, **nops** és **whattype** eljárások segítségével. Hasonlítsuk össze a 6.9. ábrán látható fákat!

Azt mondhatjuk, hogy $\frac{\sin^2 x-1}{\sin x-1}$ a $\sin x$ egész együtthatós racionális kifejezése. A Maple is így gondolja:

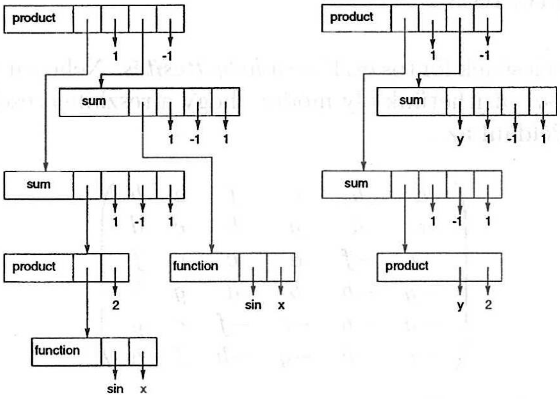
```
> type( r, ratpoly( integer, sin(x) ) );
```

true

Ez magyarázza a következő eredményt.

```
> normal( r );
```

$$\sin(x)+1$$



6.9. ábra: $\dfrac{\sin^2 x - 1}{\sin x - 1}$ és $\dfrac{y^2 - 1}{y - 1}$ reprezentációs fája

A Maple a $\dfrac{\sin^2 x - 1}{\sin x - 1}$ kifejezést *általánosított racionális kifejezésnek* tekinti. Amikor a rendszer erre a **normal** eljárást alkalmazza, több dolog is történik. Először az összes elemi transzcendens (trigonometrikus, logaritmus, exponenciális, négyzetgyök stb.) függvény argumentumát rekurzív módon normalizálja, majd egy-egy egyértelmű névvel helyettesítve „befagyasztja". Ezután az így keletkezett racionális kifejezésre alkalmazza a **normal** eljárást. Végül a befagyasztott kifejezéseket újra „felolvasztja". A **normal** és a **factor** eljárások alkalmazása során a Maple automatikusan elvégzi a részkifejezések befagyasztását és felolvasztását. Néha azonban a **frontend** eljárással ki kell segíteni a rendszert, hogy egy általános kifejezést is racionális kifejezésként tudjon kezelni:

```
> num := numer(r); den := denom(r);
```

$$num := \sin(x)^2 - 1$$

$$den := \sin(x) - 1$$

```
> gcd( num, den );

Error, (in gcd)
arguments must be polynomials over the rationals

> frontend( gcd, [num,den] );
```

$$\sin(x) - 1$$

A **gcd** eljárás a racionális számok fölötti polinomokat vár. A **frontend**-et arra használtuk, hogy időlegesen befagyasszuk (egyedi névvel helyettesítsük) a `sin(x)`-et a num és a den kifejezésekben. Az így kapott eredményre alkalmazhatjuk a **gcd** eljárást. Végül a befagyasztott neveket felolvasztja a rendszer. A következő fejezetben megismerkedünk a részkifejezések befagyasztásának egy másik módszerével.

6.3. Helyettesítés

A kifejezések kezelésének fontos eszköze a *helyettesítés*. Nehezen olvasható, nagy kifejezéseket egyszerűsíthetünk oly módon, hogy a részkifejezéseket kisebbekkel helyettesítjük. Például az

$$\begin{pmatrix} a & b & e & f & a & b \\ c & d & g & h & c & d \\ -e & -f & a & c & e & f \\ -g & -h & b & d & g & h \\ -a & -b & -e & -f & e & g \\ -c & -d & -g & -h & f & h \end{pmatrix}$$

mátrix blokkos alakban a következő módon írható:

$$\begin{pmatrix} X & Y & X \\ -Y & X^T & Y \\ -X & -Y & Y^T \end{pmatrix}$$

ahol

$$X = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{és} \quad y = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

Sok esetben lényegesen kényelmesebb az eredeti helyett a blokkos mátrix használata.

Hosszú formulák kiírásakor a Maple rövidítéseket vezet be a nagyobb közös részkifejezésekre; ezeket rendre a %1, %2, ... cimkékkel jelöli.

Helyettesítések elvégzésére rendszerint a **subs** eljárást használjuk. Ennek legegyszerűbb formája a következő:

subs(*var* = *replacement*, *expression*).

Hatására a *var* változó összes előfordulása a megadott *replacement*-tel helyettesítődik az *expression* kifejezésben. Nézzünk egy példát.

```
> kinetic_energy := momentum^2 / (2*mass);
```

$$kinetic_energy := \frac{1}{2}\,\frac{momentum^2}{mass}$$

```
> subs( momentum = mass * velocity, kinetic_energy );
```

$$\frac{1}{2}\,mass\;velocity^2$$

```
> kinetic_energy;
```

$$\frac{1}{2}\,\frac{momentum^2}{mass}$$

Figyeljük meg, hogy a kinetic_energy változó a helyettesítés után is megtartja eredeti értékét. Ezt csak értékadással lehet megváltoztatni.

```
> kinetic_energy := subs( momentum = mass * velocity,
>   kinetic_energy ):
> kinetic_energy;
```

$$\frac{1}{2}\,mass\;velocity^2$$

A következő két példa azt mutatja, hogy a helyettesítés eredményét a rendszer egyszerűsíti, de nem értékeli ki. Erről magunknak kell gondoskodnunk.

```
> # substitution but no evaluation
> subs( x=0, cos(x) * ( sin(x) + x^2 + 1 ) );
```

$$\cos(0)\,(\sin(0)+1)$$

```
> eval("); # extra evaluation
```

$$1$$

```
> sum( 1 / binomial(n,k), k=1..n );
```

$$\sum_{k=1}^{n}\frac{1}{\mathrm{binomial}(n,\,k)}$$

```
> subs(n=3,") = eval( subs(n=3,") );
```

$$\sum_{k=1}^{3}\frac{1}{\mathrm{binomial}(3,\,k)}=\frac{5}{3}$$

Ha azt akarjuk, hogy a kiértékelés automatikusan megtörténjen a helyettesítés után, elegendő a következő parancsot beírni (elhelyezhetjük az inicializáló fájlunkban is):

```
> macro( subs = eval @ subs ):
```

Erre vonatkozóan lásd a 8.7. alfejezetet is.

A többszörös helyettesítésre példaként tekintsük a Maple-ben a következő módon leírható kettős helyettesítést:

subs(*var1* = *replacement1*, *var2* = *replacement2*, *expression*).

Ezen kettős helyettesítés hatására először a *var1* változó összes előfordulása *replacement1*-gyel helyettesítődik az *expression* kifejezésben, majd az így kapott közbülső eredményben *var2* összes előfordulását cseréli ki *replacement2*-vel a rendszer. Általánosan is igaz az, hogy többszörös helyettesítés esetén a megadott helyettesítéseket balról jobbra haladva alkalmazza a Maple a megfelelő részeredményekre. A helyettesítésnek ezt a formáját *szekvenciális helyettesítésnek* nevezzük. Két példa:

```
> kinetic_energy := momentum^2 / (2*mass);
```

$$kinetic_energy := \frac{1}{2}\frac{momentum^2}{mass}$$

```
> subs( momentum = mass * velocity,
>   velocity = acceleration * time, kinetic_energy );
```

$$\frac{1}{2}\, mass\ acceleration^2\ time^2$$

```
> expression := 1 + tan(x)^2;
```

$$expression := 1 + \tan(x)^2$$

```
> subs( tan(x) = sin(x)/cos(x), sin(x)^2 = 1 - cos(x)^2,
>   expression );
```

$$1 + \frac{1 - \cos(x)^2}{\cos(x)^2}$$

```
> normal(");
```

$$\frac{1}{\cos(x)^2}$$

A szekvenciális helyettesítésen kívül úgynevezett *szimultán helyettesítést* is végezhetünk. Ehhez csupán a helyettesítéseket leíró egyenleteket kell egy halmazba vagy egy listába összefoglalni:

subs($\{var1 = replacement1,\ var2 = replacement2\}$, *expression*).

A szimultán helyettesítésnél az *expression* kifejezésben az összes helyettesítés egyszerre történik. Pontosabban kifejezve a Maple bejárja a kifejezést reprezentáló fát, és minden egyes részkifejezést balról jobbra haladva összehasonlít a helyettesítéseket definiáló egyenletek baloldalával. Ha valahol egyezést talál, akkor az eredeti kifejezésben az adott részkifejezést a megfelelő egyenlet jobb oldalával helyettesíti. A következő példák jobban érzékeltetik a szekvenciális és a szimultán helyettesítés közti különbséget:

```
> subs( x=y, y=z , x*y^2 ); # sequential substitution
```

$$z^3$$

```
> subs( { x=y, y=z }, x*y^2 ); # simultaneous substitution
```

$$y\, z^2$$

```
> subs( a=b, b=c, c=a, a + 2*b + 3*c );
```

$$6\, a$$

```
> subs( { a=b, b=c, c=a }, a + 2*b + 3*c );
```

$$b + 2\, c + 3\, a$$

```
> subs({ p=q, q=p }, f(p,q) ); # interchange p and q
```

$$f(q, p)$$

```
> [a+b=d,a=b^2,b=c^2];
```

$$[a + b = d,\ a = b^2,\ b = c^2]$$

```
> subs( ", (a+b)^2*b*a );
```

$$d^2\, c^2\, b^2$$

A **subs** eljárás nem csak változókat, hanem nagyobb részkifejezéseket is képes helyettesíteni. A helyettesítés azokra a részkifejezésekre korlátozódik, melyeket a rendszer fölismer. Ez azt jelenti, hogy csak azok a részkifejezések helyettesíthetők, amelyeket az **op** eljárás (ismételt) alkalmazásával el tudunk érni. Például:

```
> expr1 := x*y + z; expr2 := x*y*z; expr3 := (x*y)^2;
```

$$expr1 := x\,y + z$$

$$expr2 := x\,y\,z$$

$$expr3 := x^2\,y^2$$

```
> subs( x*y = w, expr1 );
```

$$w + z$$

```
> subs( x*y = w, expr2 );
```

$$x\,y\,z$$

```
> subs( x*y = w, expr3 );
```

$$x^2\,y^2$$

```
> op( expr1 );
```

$$x\,y,\ z$$

```
> op( expr2 );
```

$$x,\ y,\ z$$

```
> op( expr3 );
```

$$x^2,\ y^2$$

Mivel a második és a harmadik kifejezésben előforduló x*y szorzatot a Maple nem tekinti részkifejezésnek, közvetlenül nem is helyettesíthető ez a szorzat. Ha mégis erre van szükségünk, akkor például a következő módon járhatunk el:

```
> subs( x = w / y, expr2 );
```

$$w\,z$$

```
> subs( {x = w, y = 1}, expr2 );
```

$$w\,z$$

```
> subs( x*y*z = product*z, expr2 );
```

$$product\ z$$

Az utolsó helyettesítésnek csak akkor van értelme, ha expr2 egy x-et tartalmazó nagyobb formula része, s a többi x-et tartalmazó részkifejezésnek változatlanul kell maradnia. Teljesen hasonlóan

```
> expression := a + b + c;
```

$$expression := a + b + c$$

```
> subs( a + b = d, expression );
```

$$a + b + c$$

```
> subs( a = d - b, expression );
```
$$d+c$$
```
> subs( {a = d, b = 0}, expression );
```
$$d+c$$
```
> subs( a + b + c = d + c, expression );
```
$$d+c$$

A Maple rendszer tartalmazza az **algsubs** (**alg**ebraic **subs**titution) eljárást, melyet „matematikai" helyettesítések végrehajtására terveztek. (Ez nem csak szintaktikus helyettesítésen alapul, mint a **subs**.) A fenti példára alkalmazva a következőt kapjuk:

```
> algsubs( a + b = d, expression );
```
$$d+c$$

Az algebrai helyettesítés nem mindig egyértelmű. Mi legyen például

$$\texttt{algsubs}(a+b=d, a+2*b+3*c)$$

eredménye?

$$b+d+3*c$$

vagy

$$-a+2*c+2*d?$$

És ha mindkét válasz lehetséges, hogyan oldható ez föl? A problémára részleges megoldást ad az **algsubs** két opcionális argunemtuma, melyekkel előírhatjuk

- a változók rendezését,
- az `exact` és a `remainder` opciók közül pontosan egyet.

A változók rendezésével arról informálhatjuk a Maple-t, hogy mely változókat szeretnénk leginkább látni az eredményben. A `remainder` módban (ez az alapföltételezés) a minták előfordulásainak megkeresése általánosított maradékos polinomosztás segítségével történik. Az `exact` módban, ha a minta összeg mondjuk $p_1 + p_2$, akkor egy részkifejezésben, például $t_1 + t_2$-ben akkor és csak akkor hajtódik végre a helyettesítés, ha $f_1 + f_2 = c \times (t_1 + t_2)$ valamely c együtthatóval. Figyeljük meg, hogy polinomosztásról beszéltünk; az **algsub** csak egész kitevős egytagokból álló mintákkal és kifejezésekkel működik. Vagyis $x^{\frac{1}{2}}$ alakú mintákat és kifejezéseket az **algsub** nem vesz figyelembe. Példákkal világítjuk meg a helyzetet:

```
> algsubs( a + b = d, a + 2*b + 3*c );
```
$$d+b+3c$$
```
> algsubs( a + b = d, a + 2*b + 3*c, 'exact' );
```
$$a+2b+3c$$
```
> algsubs( a + b = d, a + 2*b + 3*c, [a,b] ); # a<b
```
$$d+b+3c$$

```
> algsubs( a + b = d, a + 2*b + 3*c, [b,a] ); # b<a
```

$$-a+2d+3c$$

```
> algsubs( a^2 + 1 = d, (a^2+1)^4 + a^3 + 2*a^2 + 1 );
```

$$d^4+(-1+d)a-1+2d$$

```
> algsubs( a^2 + 1 = d, (a^2+1)^4 + a^3 + 2*a^2 + 1,
>    'exact' );
```

$$d^4+a^3+2a^2+1$$

Figyeljük meg a különbséget:

```
> subs( a^2 = d - 1, (a^2+1)^4 + a^3 + 2*a^2 + 1 );
```

$$d^4+a^3-1+2d$$

Az **algsubs** egyik fontos alkalmazása a hatványsorok csonkítása:

```
> sum( x^k, k=-5..5 );
```

$$\frac{1}{x^5}+\frac{1}{x^4}+\frac{1}{x^3}+\frac{1}{x^2}+\frac{1}{x}+1+x+x^2+x^3+x^4+x^5$$

```
> algsubs( x^3 = 0, " );
```

$$\frac{1}{x^5}+\frac{1}{x^4}+\frac{1}{x^3}+\frac{1}{x^2}+\frac{1}{x}+1+x+x^2$$

```
> algsubs( 1/x^3 = 0, " );
```

$$1+\frac{1}{x^2}+\frac{1}{x}+x+x^2$$

Az **algsubs** általánosított racionális kifejezésekre is működik:

```
> algsubs( sin(x)^2 + cos(x)^2 = 1,
>    sin(x)^2 / ( sin(x)^3 + cos(x)^3 ) );
```

$$\frac{\sin(x)^2}{-\sin(x)^2\cos(x)+\cos(x)+\sin(x)^3}$$

```
> algsubs( sin(x)^2 + cos(x)^2 = 1,
>    sin(x)^2 / ( sin(x)^3 + cos(x)^3 ), [cos(x),sin(x)] );
```

$$\frac{\sin(x)^2}{-\sin(x)^2\cos(x)+\cos(x)+\sin(x)^3}$$

Az **algsubs** technikailag némileg különbözik a **subs**-tól:

- nem végezhetünk egy parancsban szekvenciális vagy szimultán helyettesítést,
- a helyettesítés után automatikus kiértékelés történik,
- az eljárásneveket és az indexeket nem kezeli.

Zárópéldánk a polinomokkal végzett számolással kapcsolatos:

```
> algsubs( x*y^2 = s, x^3*y^4 );
```

$$s^2x$$

```
> algsubs( x*y^2 = x*y, x^3*y^4 );
```

$$x^3\, y^2$$

Itt azt láthatjuk, hogy a helyettesítő kifejezés tartalmazhat olyan változókat, amelyek a mintában is előfordulnak. Példánkban az x^3y^4 kifejezés fölírható $x(xy^2)^2$ alakban is, ezzel magyarázható a helyettesítéssel kapott eredmény. Implicit módon a Gröbner bázisokat realizáló `grobner` csomagot is fölhasználhatjuk, ha a **simplify** eljárást mellékföltételekkel alkalmazzuk:

```
> simplify( x^3*y^2, {x*y=s}, [y,x,s] );
```

$$s^2\, x$$

```
> simplify( x^3*y^4, {x*y^2=x*y}, [y,x,s] );
```

$$x^3\, y$$

Az utolsó példa azt mutatja, hogy a mellékföltételt a rendszer egészében tekinti, azaz a jobb oldal az alkalmazott egyszerűsítési szabály része lesz. A mellékföltételekkel megadott egyszerűsítésre a 14.7. alfejezetben térünk vissza.

Mikor a Maple azt ellenőrzi, hogy egy bizonyos kifejezés előfordul-e egy szimbolikus kifejezés részkifejezéseként, a számítógépes algebrai rendszer bejárja a teljes reprezentációs fát. A helyettesítés a Maple-ben globális jellegű: a helyettesítendő részkifejezés *minden egyes* előfordulása helyettesítődik az eredeti kifejezésben, illetve a részeredményekben:

```
> expression := (x+y)^2 + x;
```

$$expression := (x+y)^2 + x$$

```
> op( expression );
```

$$(x+y)^2,\, x$$

```
> op( op( 1, (x+y)^2 ) );
```

$$x,\, y$$

```
> subs( x = z, expression );
```

$$(z+y)^2 + z$$

A helyettesítés esetén tehát nagyon fontos tudni, hogy a Maple hogyan építi fel a kifejezéseket, és mely részkifejezéseket képes a rendszer fölismerni.

Ne felejtsük el, hogy a Maple a közös részkifejezéseket csak egyszer tárolja. A **subs** eljárás végrehajtásakor a megfelelő részkifejezés minden előfordulása helyettesítésre kerül. Van azonban egy másik lehetőség: a **subsop** (**subs**titute **op**erands) eljárás segítségével a kifejezés egy vagy több operandusát helyettesíthetjük. Például a

subsop(*num1* = *replacement1*, *num2* = *replacement2*, *expression*)

utasítás hatására az *expression* kifejezés **op**(*num1*, *expression*) és **op**(*num2*, *expression*) részkifejezései egyidejűleg helyettesítődnek a *replacement1*-gyel, illetve a *replacement2*-vel. A

subsop([*i,j*] = *replacement*, *expression*)

hatására a megadott kifejezés i-dik operanduszának j-dik operandusza helyettesítődik az általunk specifikált *replacement*-tel. Néhány példával illusztráljuk, hogyan működik mindez:

```
> expression := x^2 + x + 1/x;
```

$$expression := x^2 + x + \frac{1}{x}$$

```
> subsop( 3 = y, expression ); # replace 3rd operand
```

$$x^2 + x + y$$

```
> subsop( 3 = 0, expression ); # replace 3rd operand
```

$$x^2 + x$$

```
> subsop( 1 = z, 2 = y, expression );
```

$$z + y + \frac{1}{x}$$

```
> subsop( [3,1] = y, expression );
```

$$x^2 + x + \frac{1}{y}$$

```
> 1 + 1/(1+cos(x^2+1/x^2));
```

$$1 + \frac{1}{1 + \cos(x^2 + \frac{1}{x^2})}$$

```
> subsop( [2,1,2,1,2] = y, " );
```

$$1 + \frac{1}{1 + \cos(x^2 + y)}$$

```
> subsop( 0 = J, BesselJ(1,x) );
```

$$\mathrm{J}(1, x)$$

A példák jól mutatják a **subsop** eljárás lokális jellegét. Csak a kifejezés megadott részeiben történik helyettesítés, minden más operandus érintetlen marad. Ez éppen az ellenkezője a globális jellegű **subs** eljárásnak, amellyel egy vagy több részkifejezés összes előfordulását helyettesítjük, függetlenül attól, hogy ezek hol fordulnak elő az adott kifejezésben. Ha csak a szimbolikus kifejezés valamely részével kell dolgoznunk, akkor jól ki tudjuk használni a **subsop** lokális viselkedését. Az eljárás másik előnye, hogy nem kell újragépelnünk az esetleg igen nagy helyettesítendő részkifejezést. Két példa:

```
> poly:= ( x^2 + y^2 + 2*x*y ) * ( (x+y)^2 + 1);
```

$$poly := (x^2 + y^2 + 2\,x\,y)\,((x + y)^2 + 1)$$

```
> factor( poly );
```

$$(x + y)^2\,(x^2 + y^2 + 2\,x\,y + 1)$$

```
> subsop( 1 = factor( op(1, poly) ), poly );
```

$$(x + y)^2\,((x + y)^2 + 1)$$

```
> expression := (x^2 + 2*x + 1)^2 + (x^2 - 2*x +1)^2;
```

$$expression := (x^2 + 2\,x + 1)^2 + (x^2 - 2\,x + 1)^2$$

```
> factor( expression );
```

$$2\,x^4 + 12\,x^2 + 2$$

```
> subsop( 1 = factor( op(1,expression) ),
>   2 = factor( op(2,expression) ), expression );
```

$$(x + 1)^4 + (x - 1)^4$$

Mivel gyakran van szükség kifejezések operanduszain végzett hasonló műveletekre, két könyvtári függvénnyel, az **applyop**-pal és a **map**-pel tehetjük könnyebbé életünket. Ezek működését szintén az előző példákon mutatjuk be:

```
> applyop( factor, 1, poly );
```

$$(x + y)^2\,((x + y)^2 + 1)$$

Valóban, az

applyop(*func*, *index*, *expression*)

parancs hatása ugyanaz, mintha ezt írtuk volna:

subsop(*index* = *func*(**op**(*index*, *expression*)), *expression*).

A második példában a kifejezés első szintjén található összes operanduszra akarunk egy függvényt alkalmazni. Erre a feladatra a **map** eljárás a legalkalmasabb.

```
> map( factor, expression );
```

$$(x + 1)^4 + (x - 1)^4$$

A **map**(*procedure*, *expression*) parancs hatása a következő: a *procedure* eljárást alkalmazzuk külön-külön az *expression* kifejezés minden egyes operanduszára, majd az eredményekből az eredetivel azonos típusú kifejezést állítunk össze, végül automatikus egyszerűsítést hajtunk végre (amely természetesen még megváltoztathatja a kifejezés tipusát). A 8.8. alfejezetben még visszatérünk a **map** eljárásra.

Jegyezzük meg, hogy az első példában a helyettesítést végrehajthattuk volna a következő módon is:

```
> subs( x+y = z, poly );
```

$$(x^2 + y^2 + 2\,x\,y)\,(z^2 + 1)$$

```
> factor(");
```

$$(x + y)^2\,(z^2 + 1)$$

```
> subs( z=x+y, " );
```

$$(x + y)^2\,((x + y)^2 + 1)$$

A részkifejezések időleges helyettesítésének technikája, vagyis amikor egy részkifejezést egy változóval helyettesítünk, a részeredményekkel tovább számolunk, majd visszahelyettesítjük a „befagyasztott" részkifejezést, gyakran használatos egyszerűsítési stratégia. Most ennek a módszernek újabb illusztrációja következik:

```
> expression := (x+y)^2 + 1/(x+y)^2;
```

$$expression := (x+y)^2 + \frac{1}{(x+y)^2}$$

Az $(x+y)^2$ hatvány kifejtése nélkül szeretnénk ezt a kifejezést $\frac{numerator}{denominator}$ alakra hozni. A **normal** eljárás nem egészen azt adja, amit mi szeretnénk.

```
> normal( expression );
```

$$\frac{x^4 + 6x^2y^2 + 4x^3y + 4xy^3 + y^4 + 1}{(x+y)^2}$$

Ha azonban az $x + y$ kifejezést időlegesen egy új változóval, mondjuk z-vel helyettesítjük, normalizáljuk az így kapott racionális függvényt, és végül visszahelyettesítjük az $x + y$-t z helyére, akkor a kívánt eredményt nyerjük:

```
> subs( x + y = z, expression );
```

$$z^2 + \frac{1}{z^2}$$

```
> normal(");
```

$$\frac{z^4+1}{z^2}$$

```
> subs( z = x + y, " );
```

$$\frac{(x+y)^4+1}{(x+y)^2}$$

Mivel egy hosszú Maple szekció során nem mindig könnyű visszaemlékezni arra, hogy milyen változókat használtunk és milyen helyettesítéseket alkalmaztunk, sokszor jól jönnek a Maple rendszer **freeze** és **thaw** segédfüggvényei. Amikor ezeket használjuk, a Maple maga választ új neveket a részkifejezéseknek, és nyomon is követi felhasználásukat. A fenti példa így is megoldható:

```
> readlib( freeze ):  # load the library function
> subs( x+y=freeze(x+y), expression );
```

$$freeze/R0^2 + \frac{1}{freeze/R0^2}$$

```
> normal(");
```

$$\frac{freeze/R0^4 + 1}{freeze/R0^2}$$

```
> thaw(");
```

$$\frac{(x+y)^4+1}{(x+y)^2}$$

Ezt az alfejezetet néhány olyan egyszerű példával zárjuk, melyek a specializálás problémáit illusztrálják:

```
> integrate( 1/(x-a)^2, x = 0..2 );
```

$$\frac{1}{-2+a}-\frac{1}{a}$$

A Maple sikeresen megoldott egy általános problémát (az a paraméternek még nem adtunk értéket). Ha azonban specializáljuk a-t, furcsa eredményekre jutunk. Az integrálba $a = 1$-et helyettesítve pozitív integrandus mellett negatív integrálértéket kapunk!

```
> subs( a = 1, " );
```

$$-2$$

Ha közvetlenül az $\int_0^2 \frac{1}{(x-1)^2}\,dx$ integrálra kérdezünk rá, a Maple megadja a helyes választ.

```
> integrate( 1/(x-1)^2, x = 0..2 );
```

$$\infty$$

Sajnos, nem segít a Maple-ben annak föltételezése, hogy a 0 és 2 között van.

Tekintsük a következő a-tól függő paraméteres egyenletet.

```
> eq := (a^2-1) * x^2 + (2*a^2+a-3) * x + 2*a - 2 = 0;
```

$$eq := (a^2-1)\,x^2+(2\,a^2+a-3)\,x+2\,a-2=0$$

Az $a = 1$ esetben triviális egyenletet kapunk.

```
> subs( a = 1, " );
```

$$0=0$$

Az általános esetet megoldva, majd $a = 1$-et helyettesítve csak két megoldást kapunk.

```
> solve( eq, x );
```

$$-\frac{1}{a+1},\ -2$$

```
> subs( a = 1, {"} );
```

$$\{-2, \frac{-1}{2}\}$$

Határozzuk meg az $M = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$ mátrix Jordan-féle normálformáját.

```
> with(linalg):

Warning, new definition for norm

Warning, new definition for trace
> A := matrix([[1,a],[0,1]]);
```

$$A := \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

```
> jordan(");
```

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Ez az eredmény nyilvánvalóan nem helyes, ha $a = 0$.

```
> B := matrix([[a,1],[0,1]]);
```

$$B := \begin{bmatrix} a & 1 \\ 0 & 1 \end{bmatrix}$$

```
> jordan(");
```

$$\begin{bmatrix} 1 & 0 \\ 0 & a \end{bmatrix}$$

Az előző eredmény nem érvényes az $a = 1$ esetben.

6.4. Gyakorlatok

1. Írjuk le részletesen a $2x(y^2+1)^2$ polinom belső reprezentációját. Rajzoljuk föl a megfelelő DAG-ot és a reprezentációs fát. Írjuk le mindazokat a részkifejezéseket, amelyeket a Maple fölismer. Válaszunkat támasszuk alá a **whattype**, **nops** és az **op** eljárások alkalmazásával. A **dismantle**, **addressof**, **pointto**, **disassemble** és **assemble** utasítások segítségével részletesen ellenőrizhetjük válaszunkat.

2. Transzformáljuk át az $(x+y)^2 + \dfrac{1}{x+y}$ kifejezést az $\dfrac{(x+y)^3+1}{x+y}$ kifejezésbe és fordítva.

3. Transzformáljuk át az

$$x^2 + 2\,x + 1 + \frac{1}{x^2 + 2\,x + 1}$$

kifejezést az

$$\frac{(x+1)^4 + 1}{(x+1)^2}$$

kifejezésbe és fordítva.

4. Magyarázzuk meg a következő helyettesítés eredményét:

```
> x^2-x+1/x-1/x^2;
```

$$x^2 - x + \frac{1}{x} - \frac{1}{x^2}$$

```
> subs(-1=1,");
```

$$x^2+2\,x+\frac{1}{x^2}$$

5. Hozzuk az

$$\frac{(x+1)^{10}-2\,y}{(x+y)^{10}}+\frac{1}{(x+y)^9}-\frac{x}{(x+y)^{10}}$$

kifejezést az

$$\frac{(x+1)^{10}-y}{(x+y)^{10}}$$

alakra.

7.

Polinomok és racionális kifejezések kezelése

Ez a fejezet a polinomok és a racionális kifejezések kezelésének szisztematikus ismertetését tartalmazza. A következő műveleteket vizsgáljuk: polinomok és racionális kifejezések kifejtése, szorzattá alakítása, normalizálása, összegyűjtése és rendezése. A racionális számok fölötti kifejezések kezelése mellett más tartományok (véges testek, algebrai számok és algebrai függvénytestek) fölötti kifejezéseket is tekintünk. Továbbá rövid elméleti bevezetőt nyújtunk a kanonikus és a normálalak szerinti egyszerűsítésekhez.

7.1. Kifejtés

Tételezzük föl, hogy az $(x^2-x)(x^2+2x+1)$ polinomot faktorizált alakjából az $x^4+x^3-x^2-x$ kifejtett kanonikus alakjára kívánjuk transzformálni. Használhatjuk az **expand** eljárást, szükség esetén az eredményt a **sort**-tal rendezhetjük. Az **expand** először összeszorozza a tényezőket, majd második lépésként összevonja a hasonló tagokat. Példánkban az első lépés az $x^4+2x^3+x^2-x^3-2x^2-x$ kifejezéshez vezet, a rendszer ezt az $x^4+x^3-x^2-x$ polinommá egyszerűsíti. (Ha már korábban is előfordult, s emiatt a memóriában megtalálható ez a polinom, esetleg más lesz a tagok sorrendje; ezen a **sort** eljárással segíthetünk.)

```
> partially_factored_form:= (x^2-x) * (x^2+2*x+1);
```

$$partially_factored_form := (x^2-x)\,(x^2+2\,x+1)$$

```
> expanded_form := expand( partially_factored_form );
```

$$expanded_form := x^4 + x^3 - x^2 - x$$

A Maple az összegek szorzataira a disztributív azonosságokat alkalmazza. Ez legjobban egy egyszerű példával szemléltethető:

```
> factored_form := (a+b) * (c+d);
```

$$factored_form := (a + b)(c + d)$$

```
> expanded_form := expand( factored_form );
```

$$expanded_form := a\,c + a\,d + b\,c + b\,d$$

A Maple **expand** eljárása elvégzi az összegek összeszorzását, ezután a Maple egyszerűsítő eljárása automatikusan összegyűjti a hasonló tagokat. Ezeket a lépéseket együttesen alkalmasabb a *teljes kifejtés* szakkifejezéssel illetni. Utolsó példánk esetében a szorzatalak kifejtését a következő kétlépéses mechanizmussal írhatjuk le: először a szorzatalak az $a(c + d) + b(c + d)$ alakba transzformálódik, majd a rendszer ezt a köztes kifejezést a végső $ac + ad + bc + bd$ alakra transzformálja át. Ha el akarjuk kerülni ezt a második lépést, akkor az **expand** eljárás hívásakor kell specifikálnunk, hogy a $c + d$ részkifejezést változatlanul akarjuk hagyni:

```
> partially_expanded_form := expand( factored_form, c+d );
```

$$partially_expanded_form := (c + d)\,a + (c + d)\,b$$

Az **expand** eljárás összegek egészkitevős hatványait is kifejti oly módon, hogy ezen összeg ismételt szorzatának tekinti az egészkitevős hatványt:

```
> power := (x+1)^3;
```

$$power := (x + 1)^3$$

```
> expand(");
```

$$x^3 + 3\,x^2 + 3\,x + 1$$

A negatív kitevős hatványokat az **expand** érintetlenül hagyja:

```
> power := (x+1)^(-2);
```

$$power := \frac{1}{(x + 1)^2}$$

```
> whattype(");
```

$$\hat{}$$

```
> op( power );
```

$$x + 1,\ -2$$

```
> expand( power );
```

$$\frac{1}{(x + 1)^2}$$

Ha az utóbbi kifejezést $(x + 1)^2$ nevezőjű racionális kifejezésnek tekintjük, akkor a nevezőt külön ki tudjuk fejteni:

```
> 1 / expand( denom(") );
```

$$\frac{1}{x^2 + 2\,x + 1}$$

A Maple nem fejti ki az egésztől különböző kitevős hatványokat sem:

```
> power := (x+1)^(3/2);
```

$$power := (x+1)^{3/2}$$

```
> expand(");
```

$$(x+1)^{3/2}$$

Racionális kifejezések esetében az **expand** hatására csupán a számlálóban lévő összegek fejtődnek ki:

```
> (x+1)^2 / ((x^2+x)*x);
```

$$\frac{(x+1)^2}{(x^2+x)\,x}$$

```
> expand(");
```

$$\frac{x}{x^2+x}+\frac{2}{x^2+x}+\frac{1}{(x^2+x)\,x}$$

Eddig csak az egészek fölött definiált polinomok és racionális függvények kifejtését tekintettük. Ha véges gyűrűk, algebrai számok vagy algebrai függvénytestek fölött definiált polinomokkal dolgozunk, akkor az **expand** eljárás **Expand** nevű tétlen formáját kell használnunk, szükség esetén az **evala** (**eval**uate in a context of **a**lgebraic numbers or function fields) eljárással kombinálva. Nézzünk erre néhány példát.

- Kifejtés $\mathbb{Z}_8$ fölött.

```
> Expand( (x+1)^8 ) mod 8;
```

$$x^8+4\,x^6+6\,x^4+4\,x^2+1$$

- Kifejtés $\mathbb{Q}_\alpha$ fölött, ahol α a z^5+z+1 polinom gyöke.

```
> alias(alpha = RootOf( z^5 + z + 1, z)):
> (x+alpha)^5;
```

$$(x+\alpha)^5$$

```
> evala( Expand(") );
```

$$x^5+5\,\alpha\,x^4+10\,\alpha^2\,x^3+10\,\alpha^3\,x^2+5\,\alpha^4\,x-\alpha-1$$

- Kifejtés $\mathbb{Z}_5(\alpha)$ fölött, ahol α a z^5+z+1 polinom gyöke.

```
> Expand("") mod 5;
```

$$x^5+4\,\alpha+4$$

- Kifejtés a $\mathbb{Q}(\sqrt{1+y})$ algebrai függvénytest fölött.

  Emlékeztetünk arra, hogy $\mathbb{Q}$ fölötti algebrai függvényen olyan függvényt értünk, amely anullál valamely racionális függvénytestből vett együtthatós egyváltozós polinomot. Példánkban $\sqrt{1+y}$ a $\mathbb{Q}(y)$ racionális függvénytestből vett együtthatókkal fölírt z^2-1-y polinom gyökeként van definiálva.

```
> alias( beta = RootOf( z^2 - 1 - y, z ) ):
> (x+beta)^2;
```

$$(x+\beta)^2$$

```
> evala( Expand(") );
```

$$x^2+2\beta x+1+y$$

7.2. Szorzattá alakítás

A **factor** Maple eljárás az **expand** „nagy testvére”. A polinom $\mathbb{Q}$ fölött irreducibilis tényezőkkel fölírt szorzatalakját számítja ki. A kapott eredmény, amelyet *faktorizált normálformának* nevezünk, a szorzótényezők sorrendjétől eltekintve egyértelmű. Az eljárás racionális kifejezésekre is alkalmazható, ekkor a számláló és a nevező faktorizálása előtt a közös tényezőket törli a rendszer.

Az 5.1. alfejezetben már láttuk, hogy a Maple képes különböző tartományok fölötti polinomok faktorizálására. Most további példákat mutatunk be. Véges testek és algebrai függvénytestek fölötti polinomok esetében szükség lehet a **Factor** tétlen forma használatára. A Maple véges testek fölötti egyváltozós polinomok esetében Cantor és Zassenhaus [33], valamint Berlekamp [16] algoritmusát használja, többváltozós polinomokra a [133, 202] szerinti Hensel-féle föloldást végez. Algebrai számok és függvénytestek fölött értelmezett polinomok faktorizálására Lenstra [129] és Kronecker–Trager [182] eljárását alkalmazza.

- Faktorizálás $\mathbb{Q}(\sqrt{6})$ fölött:

```
> factor( 8*x^3 - 12*x, sqrt(6) );
```

$$2\,(2x-\sqrt{6})\,(2x+\sqrt{6})\,x$$

- Faktorizálás a $\mathbb{Z}_2, \mathbb{Z}_3$ és $\mathbb{Z}_5$ véges testek fölött:

```
> x^4 + 1;
```

$$x^4+1$$

```
> Factor(") mod 2;
```

$$(x+1)^4$$

```
> Factor("") mod 3;
```

$$(x^2+x+2)\,(x^2+2x+2)$$

```
> Factor(""") mod 5;
```

$$(x^2+3)\,(x^2+2)$$

- Faktorizálás $\mathbb{Z}_7(\alpha)$ fölött, ahol α a z^7+z^3+1 polinom gyökét jelenti:

```
> alias( alpha = RootOf( z^7 + z^3 + 1, z ) ):
> x^7 + 6*alpha^3 + 6;
```

$$x^7+6\alpha^3+6$$

```
> Factor(") mod 7;
```

$$(x+\alpha)^7$$

- Faktorizálás a $\mathbb{Q}(\sqrt{1+y})$ algebrai függvénytest fölött:

```
> alias( beta = RootOf( z^2 - 1 - y, z ) ):
> x^2 + 2*beta*x + 1 + y;
```

$$x^2 + 2\beta x + 1 + y$$

```
> factor( x^2 + 2*beta*x + 1 + y, beta );
```

$$(x + \beta)^2$$

A $8x^3 - 12x$ polinom faktorizálásánál szinte a „semmiből" bukkant föl a racionális számtest bővítése $\sqrt{6}$-tal, ami a teljes faktorizációhoz szükséges. A Maple a **split** eljárással maga is meg tudja határozni tetszőleges egyváltozós polinom fölbontási testét:

```
> readlib( split ): # load the library function
> split( 8*x^3 - 12*x, x );
```

$$8\,(x + \frac{1}{2}\operatorname{RootOf}(_Z^2 - 6))\,(x - \frac{1}{2}\operatorname{RootOf}(_Z^2 - 6))\,x$$

```
> convert( ", radical );
```

$$8\,(x + \frac{1}{2}\sqrt{6})\,(x - \frac{1}{2}\sqrt{6})\,x$$

Példa $\mathbb{Q}$ fölött irreducibilis polinomra:

```
> p := x^4 + 2*y^4;
```

$$p := x^4 + 2\,y^4$$

- Faktorizálás a $\sqrt{2}$-vel és I-vel végzett testbővítés fölött:

```
> factor( p, { sqrt(2), I } );
```

$$(x^2 - I\,y^2\sqrt{2})\,(x^2 + I\,y^2\sqrt{2})$$

- Faktorizálás $\mathbb{Z}_3$ fölött:

```
> Factor( p ) mod 3;
```

$$(2\,y + x)\,(y^2 + x^2)\,(y + x)$$

- Faktorizálás a GF(9) Galois-test (azaz $\mathbb{Z}_3$ algebrai kiterjesztése az $x^2 - 2$ irreducibilis polinommal) fölött:

```
> alias( alpha = RootOf( z^2 - 2, z ) ):
> Factor( p, alpha ) mod 3;
```

$$2\,(y + 2\,x)\,(y + \operatorname{RootOf}(_Z^2 + 1)\,x)\,(y + 2\operatorname{RootOf}(_Z^2 + 1)\,x)\,(y + x)$$

```
> convert( ", radical );
```

$$2\,(y + 2\,x)\,(y + I\,x)\,(y + 2\,I\,x)\,(y + x)$$

Az **AFactor** eljárás a komplex számtest fölötti többváltozós polinomok teljes faktorizálását számítja ki:

```
> evala( AFactor( x^3 + y^3 ) );
```

$$(x + (\text{RootOf}(_Z^2 - _Z + 1) - 1)\,y)\,(x - \text{RootOf}(_Z^2 - _Z + 1)\,y)\,(y + x)$$

```
> convert( ", radical ); # go to radical notation
```

$$(x + (-\frac{1}{2} - \frac{1}{2} I\sqrt{3})\,y)\,(x - (\frac{1}{2} - \frac{1}{2} I\sqrt{3})\,y)\,(y + x)$$

A polinomok **factor**-ral végzett faktorizálásának egyik első lépése lehet a *négyzetmentes faktorizáció* előállítása. Egy nem konstans polinom négyzetmentes faktorizációja $c\,p_1\,p_2^2\,p_3^3\ldots$ alakú szorzat, ahol c racionális konstans, $p_1, p_2, p_3, \ldots$ pedig többszörös tényezők nélküli relatív prím polinomok. A Maple rendszerben a négyzetmentes faktorizációt közvetlen konverzióval vagy az **sqrfree** eljárás hívásával számíthatjuk ki:

```
> x^4 + x^3 - x^2 - x;
```

$$x^4 + x^3 - x^2 - x$$

```
> convert( ", sqrfree );
```

$$(x - 1)\,x\,(x + 1)^2$$

```
> sqrfree("");
```

$$[1, [[x - 1, 1], [x, 1], [x + 1, 2]]]$$

Amennyiben számításainkat a $\mathbb{Z}_2$ kételemű test fölött végezzük, az utóbbi polinom négyzetmentes faktorizációja $x(x+1)^3$-nel egyenlő. Ez a Maple rendszerben a következőképpen kapható meg:

```
> Sqrfree(""") mod 2;
```

$$[1, [[x, 1], [x + 1, 3]]]$$

7.3. Kanonikus alak és normálalak

Kifejezések egyszerűsítése vagy kezelése kapcsán már találkozhattunk néhányszor a *kanonikus alak* és a *normálalak* szakkifejezésekkel. Megéri kicsit hosszabban időzni ennél a témánál, még ha ez egy elméletibb jellegű közjáték lesz is. Részletes áttekintést a [26]-ban találhatunk.

Az egyszerűsítés fő problémája abban áll, hogy egy matematikai kifejezés számos ekvivalens formában fölírható, ugyanakkor nem mindig könnyű fölismerni az ekvivalenciát. Például a harmadik Hermite-polinom $8\,x^3 - 12\,x$-szel egyenlő, de könnyen megadható négy, ezzel ekvivalens formula:

$$x(8\,x^2 - 12)$$
$$4\,x(2\,x^2 - 3)$$
$$2\,x(2\,x - \sqrt{6})(2\,x + \sqrt{6})$$
$$(2\,x)^3 - 6 \cdot (2\,x)$$

Melyik a legegyszerűbb forma? Ha azt akarjuk hangsúlyozni, hogy ez $2\,x$-nek polinomja, akkor természetesen a $2\,x(2\,x - \sqrt{6})(2\,x + \sqrt{6})$ és a $(2\,x)^3 - 6 \cdot (2\,x)$ a

megfelelő jelöltek. Ha kritériumként a tagok számát vesszük, akkor a $8x^3 - 12x$ és a $(2x)^3 - 6 \cdot (2x)$ lenne a két esélyes.

Még világosabban specifikált hasonló feladat az úgynevezett *zérus ekvivalencia probléma*, vagyis annak eldöntése, hogy egy kifejezés egyenlő-e nullával. De még ez is lehet nehéz, mint például a

$$\ln\tan(\frac{1}{2}x + \frac{1}{4}\pi) - \operatorname{arcsinh}\tan x = 0$$

kifejezés vizsgálata Ennek az egyenlőségnek az ellenőrzése nem triviális feladat.

Az egyszerűsítési problémát általánosan a következőképpen kezelhetjük. Legyen $\mathcal{E}$ szimbolikus kifejezések (például az egészek fölötti egyváltozós polinomok) egy osztálya, és legyen $\sim$ az $\mathcal{E}$-n értelmezett ekvivalenciareláció. Egy ekvivalens, de egyszerűbb kifejezés megtalálásának problémáját úgy jellemezhetjük, mint egy olyan $\mathcal{S}$: $\mathcal{E} \longrightarrow \mathcal{E}$ kiszámítható transzformáció megkeresését, amely azzal a tulajdonsággal bír, hogy minden t $\mathcal{E}$-beli kifejezésre

$$\mathcal{S}(t) \sim t \quad \text{és} \quad \mathcal{S}(t) \preceq t.$$

Itt a $\preceq$ valamilyen egyszerűsítési koncepciót jelöl: $s \prec t$ jelentése „s egyszerűbb $\mathcal{E}$-ben, mint t". Ez például a következőket jelentheti: „s-nek kevesebb tagja van, mint t-nek", „s kevesebb memóriát használ, mint t", vagy „s olvashatóbb, mint t". Azt mondjuk, hogy az s és a t kifejezés megegyezik (azonos), és ezt $s \equiv t$-vel jelöljük, ha s és t ugyanazon alapelemekből ugyanúgy épül föl, vagy még pontosabban, ha mindkettőnek ugyanaz az irányított aciklikus gráf felel meg. (Lásd az előző fejezetben az adatok belső reprezentációjának leírását.) $\mathcal{S}$-et *kanonikus egyszerűsítőnek* nevezzük $\mathcal{E}$-n, ha $\mathcal{S}$ olyan kiszámítható eljárás, hogy minden $\mathcal{E}$-beli s-re és t-re

$$\mathcal{S}(t) \sim t \text{ és } s \sim t \quad \Longrightarrow \quad \mathcal{S}(s) \equiv \mathcal{S}(t).$$

A kanonikus egyszerűsítő minden ekvivalenciaosztályból egy egyértelmű kitüntetett elemet választ ki, ezt *kanonikus alaknak* nevezzük. Egyváltozós polinomok kifejtett kanonikus alakját a következőképpen kaphatjuk meg :

(i) rekurzív módon elvégezzük az összegek összeszorzását,

(ii) összegyűjtjük az azonos kitevőjű tagokat,

(iii) elhagyjuk a fölösleges tagokat, a megmaradókat pedig a kitevők csökkenő sorrendjében rendezzük.

A kanonikus egyszerűsítéssel szemben támasztott követelmények nagyon magasak, és nem is érhetők mindig el. Az egyszerűsítés egy gyengébb formája a *normál egyszerűsítés*. Tegyük föl, hogy $\mathcal{E}$-ben létezik egy 0-val jelölt kitüntetett elem. Az $\mathcal{S}$ *normál egyszerűsítő* $\mathcal{E}$-n olyan kiszámítható eljárás, hogy minden $\mathcal{E}$-beli s-re és t-re a következők igazak:

$$\mathcal{S}(t) \sim t \text{ és } t \sim 0 \quad \Longrightarrow \quad \mathcal{S}(t) \equiv \mathcal{S}(0).$$

A t kifejezést *normál formájúnak* nevezzük $\mathcal{E}$-ben, ha $\mathcal{S}(t) \equiv t$. A normálalak tehát nem szükségképpen egyértelmű az összes ekvivalenciaosztályban (kivéve a 0-t tartalmazó osztályt).

Egyváltozós polinomok kifejtett normálformája, amelyet korábban gyűjtött formának neveztünk, a következőképpen kapható meg:

(i) rekurzív módon elvégezzük az összegek összeszorzását,

(ii) összegyűjtjük az azonos kitevőjű tagokat, végül

(iii) elhagyjuk a fölösleges tagokat.

A normál egyszerűsítés általában könnyebb, mint a kanonikus egyszerűsítés. Polinomok normálformájára további példákat szolgáltat a négyzetmentes alak és a Horner-forma (az a zárójelezett alak, amelyet gyakran használnak hatékonysági okokból).

```
> poly := expand( (x+y+z)^2*(x+y+1) );
```

$$poly := x^3 + 3\,x^2\,y + x^2 + 3\,x\,y^2 + 2\,x\,y + 2\,x^2\,z + 4\,x\,z\,y + 2\,x\,z + y^3 \\ + y^2 + 2\,y^2\,z + 2\,y\,z + z^2\,x + z^2\,y + z^2$$

```
> readlib(cost)( poly);
```

$$14\ additions + 32\ multiplications$$

```
> horner_form := convert( poly, 'horner', [x,y,z] );
```

$$horner_form := z^2 + ((2+z)\,z + (2\,z + 1 + y)\,y)\,y \\ + ((2+z)\,z + (2 + 4\,z + 3\,y)\,y + (2\,z + 1 + 3\,y + x)\,x)\,x$$

```
> cost( horner_form );
```

$$14\ additions + 13\ multiplications$$

Racionális függvények parciális törtekre bontása tekinthető a függvényre alkalmazott normál egyszerűsítőnek is.

7.4. Normalizálás

A **normal** Maple eljárás a $\mathbb{Q}$ fölötti racionális függvények normál egyszerűsítője. A racionális függvényt úgynevezett *faktorizált normálformára* alakítja. Ez *számláló/nevező* alakú, ahol a számláló és a nevező egész együtthatós relatív prím polinomok, mindkettő kifejtett polinomok szorzata, és a normál egyszerűsítés során a közös tényezők változatlanok maradnak, amennyire ez csak lehetséges:

```
> (x-1)*(x+2)/((x+1)*x) + (x-1)/(1+x)^2;
```

$$\frac{(x-1)\,(x+2)}{(x+1)\,x} + \frac{x-1}{(x+1)^2}$$

```
> normal(");
```

$$\frac{(x-1)\,(x^2+4\,x+2)}{(x+1)^2\,x}$$

```
> (x^2+x-2)/((x+1)*x)+(x-1)/(1+x)^2;
```

$$\frac{x^2+x-2}{(x+1)\,x}+\frac{x-1}{(x+1)^2}$$

```
> normal(");
```

$$\frac{x^3+3\,x^2-2\,x-2}{(x+1)^2\,x}$$

```
> normal( """, 'expanded' );
```

$$\frac{x^3+3\,x^2-2\,x-2}{x^3+2\,x^2+x}$$

Az **expanded** kulcsszó megadásával jelezhetjük, ha azt kívánjuk, hogy mind a számláló, mind a nevező kifejtett normálalakú legyen.

A racionális kifejezéseknek többféle normálalakja létezik, ezek némelyikét úgy is megkaphatjuk, hogy a **normal**, **expand** vagy **factor** eljárásokat külön-külön alkalmazzuk a számlálóra és a nevezőre. A lehetséges alternatívák:

```
> ratfunc := (x^4+x^3-4*x^2-4*x) / (x^3+x^2-x-1);
```

$$\mathit{ratfunc} := \frac{x^4+x^3-4\,x^2-4\,x}{x^3+x^2-x-1}$$

- $\dfrac{\textit{faktorizált normálalak}}{\textit{faktorizált normálalak}}$

  ```
  > factor( ratfunc );
  ```

  $$\frac{(x-2)\,(x+2)\,x}{(x-1)\,(x+1)}$$

- $\dfrac{\textit{faktorizált normálalak}}{\textit{kifejtett kanonikus alak}}$

  ```
  > factor(numer(ratfunc)) / sort(expand(denom(ratfunc)));
  ```

  $$\frac{x\,(x-2)\,(x+2)\,(x+1)}{x^3+x^2-x-1}$$

- $\dfrac{\textit{kifejtett kanonikus alak}}{\textit{faktorizált normálalak}}$

  ```
  > sort(expand(numer(ratfunc))) / factor(denom(ratfunc));
  ```

  $$\frac{x^4+x^3-4\,x^2-4\,x}{(x-1)\,(x+1)^2}$$

- $\dfrac{\textit{kifejtett kanonikus alak}}{\textit{kifejtett kanonikus alak}}$

  ```
  > sort( normal( ratfunc, 'expanded' ));
  ```

  $$\frac{x^3-4\,x}{x^2-1}$$

7.5. Összegyűjtés

A **collect** eljárást a polinomok hasonló tagjaiban szereplő együtthatók egy csoportba gyűjtésére használjuk. Az összegyűjtés különböző módozatait legjobban példákkal mutathatjuk meg.

```
> poly := expand( (x+y+z)^2 * (x+y+1) );
```

$$poly := x^3 + 3x^2y + x^2 + 3xy^2 + 2xy + 2x^2z + 4xzy + 2xz + y^3 + y^2 + 2y^2z + 2yz + z^2x + z^2y + z^2$$

Ebben a példában *osztott* vagy *kifejtett alakú* Maple polinomot láttunk. Az x, y és z ismeretlenekkel fölírt osztott egész együtthatós többváltozós polinomok halmazára a $\mathbb{Z}[x, y, z]$ jelölést használjuk.

Tekintsük a *poly* polinomot, mint z polinomját, amelynek együtthatói az x és az y változók osztott polinomjai, vagyis tekintsük a *poly*-t, mint $\mathbb{Z}[x, y][z]$ egy elemét:

```
> collect( poly, z );
```

$$(x + y + 1)z^2 + (4xy + 2x + 2x^2 + 2y^2 + 2y)z + x^3 + 3x^2y + x^2 + 3xy^2 + 2xy + y^2 + y^3$$

Most tekintsük *poly*-t, mint z polinomját, melynek együtthatói y olyan polinomjai, amelynek együtthatói x egész együtthatós polinomjai, vagyis tekintsük *poly*-t, mint a $\mathbb{Z}[x][y][z]$ egy elemét:

```
> collect( poly, [z,y], 'recursive' );
```

$$(x + y + 1)z^2 + (2y^2 + (4x + 2)y + 2x + 2x^2)z + y^3 + (3x + 1)y^2 + (3x^2 + 2x)y + x^3 + x^2$$

Pongyolán beszélve azt mondhatjuk, hogy *poly*-t a z és az y olyan polinomjának tekintettük, ahol a z határozatlant kitüntetetettnek vettük.

Természetesen *poly*-t tekinthetjük olyan kétváltozós polinomnak is, amelyben egyik változót sem tüntetjük ki, vagyis tekinthetjük a *poly*-t mint $\mathbb{Z}[x][y, z]$ elemét:

```
> collect( poly, [z,y], 'distributed' );
```

$$(3x^2 + 2x)y + (2x + 2x^2)z + x^3 + x^2 + y^3 + (3x + 1)y^2 + (4x + 2)yz + (x + 1)z^2 + 2y^2z + z^2y$$

Végezetül a **collect** eljárás további argumentumaként megadhatjuk egy olyan eljárás nevét, melyet minden egyes együtthatóra külön-külön kell alkalmazni:

```
> collect( poly, z, factor ); # recursive order by default
```

$$(x + y + 1)z^2 + 2(x + y + 1)(x + y)z + (x + y + 1)(x + y)^2$$

```
> collect( poly , [z,y], 'distributed', factor );
```

$$x(3x+2)y+2x(x+1)z+x^2(x+1)+y^3+(3x+1)y^2$$
$$+(4x+2)yz+(x+1)z^2+2y^2z+z^2y$$

A Maple-ben főleg az olvashatóság, valamint a memória-használatra és a számítási időre vonatkozó hatékonysági szempontok miatt használjuk a többváltozós polinomok gyűjtött alakját. Ha a polinom valamely változójára nézve tömör (dense), akkor a tagokat erre a változóra vonatkozóan előnyös összegyűjteni, mivel ez csökkenteni fogja a fölhasznált memóriaterületet a kifejtett formához képest. A tagok összegyűjtése egy lehetséges módja a Maple objektumokra vonatkozó méretkorlátok megkerülésének. A Maple-ben egy összegnek legföljebb 65535 (azaz $2^{16}-1$) tagja lehet, ha ezt túllépjük, a rendszer a következő üzenetet küldi:

```
System Error, object too large
```

Ha a kifejezést valamely határozatlanra vonatkozóan összegyűjtött alakban tartjuk, akkor ezáltal csökkenthetjük a kifejezésben lévő összegek maximális számát. Fönti példánkban *poly* kifejtett alakja 15 tagból áll, ugyanekkor a $\mathbb{Z}[x,y][z]$ fölötti faktorizált együtthatós gyűjtött alakja mindössze 3 tagot tartalmaz,.

A **normal**-hoz hasonlóan a **collect** eljárás alkalmazható általánosított racionális kifejezésekre is:

```
> ln(x)^3/a + ln(x)^2*x/(a^2+a) + a^2*ln(x)^2*x/(a^2+a)
>   + 2*x^2/(1+a)+a*x^2/(1+a) + a^3*ln(x)/(a^2+a)
>   + 2*ln(x)*a/(a^2+a) + ln(x)/(a^2+a) + a/(a^2+a);
```

$$\frac{\ln(x)^3}{a}+\frac{\ln(x)^2\,x}{a^2+a}+\frac{a^2\ln(x)^2\,x}{a^2+a}+2\frac{x^2}{1+a}+\frac{a\,x^2}{1+a}+\frac{a^3\ln(x)}{a^2+a}$$
$$+2\frac{\ln(x)\,a}{a^2+a}+\frac{\ln(x)}{a^2+a}+\frac{a}{a^2+a}$$

```
> collect( ", [x,ln(x)], 'distributed', normal );
```

$$\frac{1}{1+a}+\frac{(2+a)\,x^2}{1+a}+\frac{(a^3+2a+1)\ln(x)}{a(1+a)}+\frac{(1+a^2)\,x\ln(x)^2}{a(1+a)}$$
$$+\frac{\ln(x)^3}{a}$$

Figyeljük meg, hogy csak nevekre és függvényhívásokra vonatkozóan lehet gyűjtéseket végezni, polinom-kifejezésekre nem. Ha a

```
> x^5 - 5*x^4*y^2 + 10*x^3*y^4 - 10*x^2*y^6 + 5*x*y^8
>   - y^10 - 2;
```

$$x^5-5x^4y^2+10x^3y^4-10x^2y^6+5xy^8-y^{10}-2$$

kifejezés első négy tagjáról az $(x-y^2)^5$ jut eszünkbe, nem kérhetjük egyszerűen az $x-y^2$ szerinti gyűjtött alakot:

```
> collect( ", x - y^2 );

Error, (in collect) cannot collect, x-y^2
```

Ehelyett az $x - y^2 = z$ mellékfeltétel szerint egyszerűsíthetünk:

```
> siderel := {z = x - y^2};
```

$$siderel := \{z = x - y^2\}$$

```
> simplify( "", siderel, [x,y,z] );
```

$$-2 + z^5$$

```
> subs( siderel, " );
```

$$-2 + (x - y^2)^5$$

Ilyen polinomiális egyszerűsítésekre vonatkozó további példák találhatók még a 14.7. alfejezetben.

7.6. Rendezés

A **sort** eljárást arra használjuk, hogy polinomokat valamilyen alkalmas sorrendbe rendezzünk. Erről már volt szó az 5.2. alfejezetben. Itt mindössze egy olyan példát mutatunk, amelyben egy általánosított racionális függvény számlálóját és nevezőjét rendezzük:

```
> r := sort( (cos(x) - sin(x)) / (cos(x) + sin(x)),
>    [cos(x),sin(x)], 'plex' );
```

$$r := \frac{\cos(x) - \sin(x)}{\cos(x) + \sin(x)}$$

```
> sort( r, [sin(x),cos(x)], 'plex' );
```

$$\frac{-\sin(x) + \cos(x)}{\sin(x) + \cos(x)}$$

7.7. Gyakorlatok

Megjegyzés: Az alábbi gyakorlatok megoldásában a tényezők sorrendje eltérhet az itt megadottól.

1. Írjuk le az egyváltozós racionális együtthatós racionális függvények egy lehetséges kanonikus egyszerűsítőjét.

2. Tekintsük a következő racionális kifejezést:

$$\frac{x^4 + x^3 - 4x^2 - 4x}{x^4 + x^3 - x^2 - x}.$$

Hozzuk ezt a kifejezést a Maple segítségével a következő alakokra:

a) $\dfrac{x^2 - 4}{x^2 - 1}$

b) $\dfrac{(x-2)(x+2)}{x^2-1}$

3. Tekintsük a következő racionális kifejezést:

$$2\frac{x^3-yx^2-yx+y^2}{x^3-yx^2-x+y}$$

Alakítsuk át a Maple segítségével a következő kifejezésekké:

a) $2\dfrac{x^2-y}{x^2-1}$

b) $2\dfrac{x^2-y}{(x-1)(x+1)}$

c) $2-\dfrac{y-1}{x-1}+\dfrac{y-1}{x-1}$

d) $2-2\dfrac{y-1}{x^2-1}$

4. Tekintsük a $(2x^2-x)(2x^2+x)$ polinomot.
Alakítsuk át a Maple segítségével a következő polinomokká:

a) $(-1+4x^2)x^2$

b) $x^2(2x-1)(2x+1)$

c) $(2x^3+x^2)(2x-1)$.

5. Tekintsük az $(x^2+xy+x+y)(x+y)$ polinomot.
Alakítsuk át a Maple-ben a következő polinomokká:

a) $x^3+2x^2y+xy^2+x^2+2xy+y^2$

b) $(x+1)(x+y)^2$

c) $y^2+(2y+y^2)x+(1+2y)x^2+x^3$

d) $x^3+x^2+(2x^2+2x)y+(x+1)y^2$.

8. Függvények

A Maple rendszerben függvénykapcsolatot három módon definiálhatunk: képlettel, a szokásos matematikai jelöléshez hasonló nyíl operátor segítségével és eljárásként. A fejezet részletesen tárgyalja mindegyik módszert. Külön figyelmet szentelünk a rekurzívan definiált eljárásoknak és függvényeknek. Részletesen vizsgáljuk a `remember` opció használatát hatékony rekurzív függvények definiálására.

Ez a fejezet nem a programozásról szól, csak arról lesz szó, hogyan kell matematikai függvényeket definiálni a Maple rendszerben. Főleg az olyan gyakorlati kérdésekre koncentrálunk, mint például „hogyan kell függvényt definiálni”, „hogyan kell egy formulát függvénnyé alakítani” , „mikor használjunk névtelen függvényeket”.

8.1. Matematikai függvények

Az előző fejezetekben már láttunk néhány olyan általánosan használt matematikai függvényt, amely a Maple-ben is rendelkezésünkre áll. Ezek teljes listáját a **?inifcns** (help about **ini**tially known **f**un**c**tio**ns**) utasítás segítségével kaphatjuk meg. Ez a fölsorolás kevéssé ismert függvények neveit is tartalmazza, ilyen például a *Lambert-féle W függvény.* Ezt a [66]-ban található következő egyenlet megoldásaként definiálták:

```
> f(x) * exp(f(x)) = x;
```

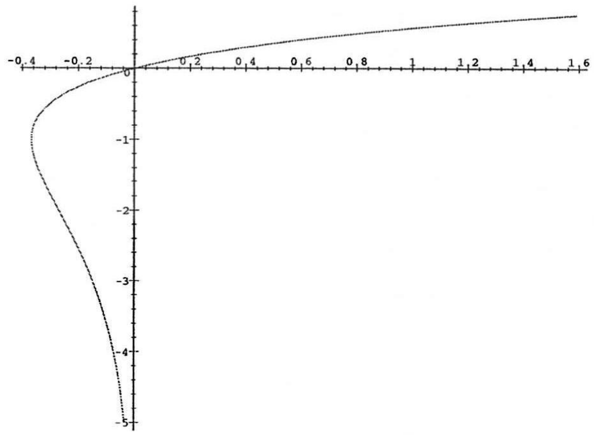
$$\mathrm{f}(x)\, e^{\mathrm{f}(x)} = x$$

```
> solve( ", f(x) );
```

$$\mathrm{LambertW}(x)$$

A Lambert-féle W függvény részletesebb leírása megtalálható [48]-ban. Ez egy többértékű komplex függvény. Két valós ágát együttesen ugyanazon az ábrán legjobban a Maple paraméteres rajzoló eljárásával jeleníthetjük meg. Az eredmény a 8.1. ábrán látható.

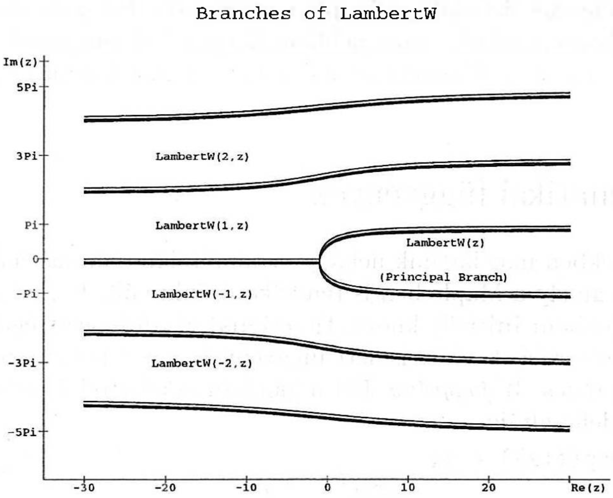
```
> plot( [ y*exp(y), y, y = -5..0.75 ] );
```



8.1. ábra: A Lambert-féle W függvény két valós ágának képe

A komplex többértékű W függvény egyes ágainak értékkészlete a **branches** eljárással jeleníthető meg. (Lásd a 8.2. ábrát.) Az eljárás mostani változata „ismeri" a trigonometrikus függvények inverzeit és a természetes alapú logaritmust is.

```
> readlib( branches ): # load the library function
> branches( LambertW, thick );
```



8.2. ábra: A Lambert-féle W függvény ágainak értékkészlete

Az alábbiakban két példát adunk olyan számításokra, melyekben ez a függvény játszik szerepet.

```
> a*x + b^x =c;
```

$$a\,x + b^x = c$$

```
> solve(",x);
```

$$-\frac{\mathrm{LambertW}\left(\dfrac{\ln(b)\,e^{\left(\frac{\ln(b)\,c}{a}\right)}}{a}\right)a - \ln(b)\,c}{a\ln(b)}$$

```
> f(x) = solve( f(x) = x^f(x), f(x) );
```

$$\mathrm{f}(x) = -\frac{\mathrm{LambertW}(-\ln(x))}{\ln(x)}$$

Az utóbbi eredmény pontosan azt jelenti, hogy tetszőleges $x \in (0,1)$-re az

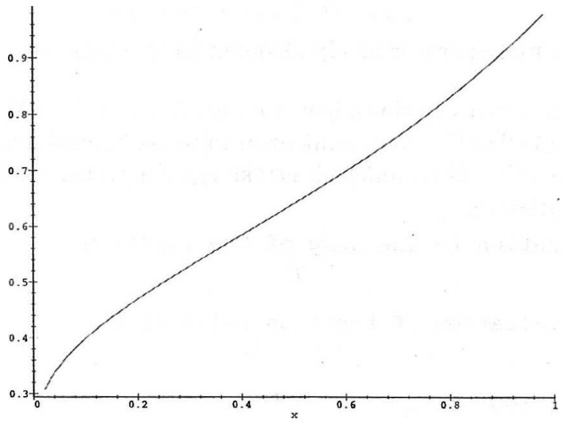
$$f : x \longmapsto x^{x^{x^{\cdot^{\cdot^{\cdot}}}}}$$

leképezés megegyezik az

$$f : x \longmapsto -\frac{\mathrm{LambertW}(-\ln x)}{\ln x}$$

függvénnyel. A függvény gráfja a 8.3. ábrán látható.

```
> plot(rhs("), x=0..1 );
```



8.3. ábra: A $-\frac{\mathrm{LambertW}(-\ln x)}{\ln x}$ függvény a $(0,1)$-en

A Maple-ben függvénykapcsolatokat három módon definiálhatunk:

- Formulával

  Valamely mennyiség változóktól való függését leírhatjuk formulával. Ha például egy test hőmérséklete az idő függvényében exponenciálisan csökken, a Maple-ben ezt a következő képlettel adhatjuk meg

```
> T := T0 * exp(-c*t);
```

$$T := T0\, e^{(-c\,t)}$$

  Ha egy adott időpontban akarjuk megkapni a hőmérsékletet, a t-t valamely konkrét értékkel kell helyettesítenünk.

```
> subs( t=1/c, T );
```

$$T0\, e^{(-1)}$$

- Függvényként

  A matematikában szokásos módon:

```
> T := t -> T0 * exp(-c*t);
```

$$T := t \to T0\, e^{(-c\,t)}$$

```
> T(2/c), T(time), T(0);
```

$$T0\, e^{(-2)},\ T0\, e^{(-c\,time)},\ T0$$

  Az úgynevezett *nyíl operátort* részletesebben a következő alfejezetben tárgyaljuk.

- Eljárásként

  Az előző függvény Maple eljárásként így írható fölé

```
> T := proc(t) T0 * exp(-c*t) end;
```

$$T := \textbf{proc}(t)\ T0 \times \exp(-c \times t)\ \textbf{end}$$

```
> T(2/c), T(time), T(0);
```

$$T0\, e^{(-2)},\ T0\, e^{(-c\,time)},\ T0$$

  Valójában a nyíl operátor az eljárásdefiníció speciális esete.

A matematikai függvények gyakran igen hasznosak lehetnek, a Maple-ben azonban gondosan meg kell különböztetnünk azon változókat, amelyek egy kifejezésre mutatnak és azon változókat, amelyek értéke egy függvény. Az előbb definiált T függvényre hivatkozva:

```
> T; # evaluation to the name of the function
```

$$T$$

```
> T(t); # evaluation of function value at t
```

$$T0\, e^{(-c\,t)}$$

```
> solve( T = 100, t );
```

Az utolsó parancs után semmilyen eredményt nem írt ki a Maple, ami azt jelzi, hogy nem talált megoldást. Ha `infolevel[solve]` értéke nagyobb nullánál, akkor szól is a rendszer, hogy nincs megoldás:

```
> infolevel[solve] := 1:
> solve( T = 100, t );
```

```
solve:    Warning: no solutions found
```

Olyan t érték valóban nem létezik, amelyre a *T függvény* a 100 számmal lenne egyenlő. Természetesen nem ezt akartuk megtudni. Ha helyesen kérdezünk, a Maple megtalálja a megoldást.

```
> solve(T(t) = 100, t );
```

$$-\frac{\ln(\frac{100}{T0})}{c}$$

8.2. A nyíl operátor

Az előző fejezetben már találkoztunk a függvények definiálásának egyik módszerével a Maple rendszerben, megismertük a *nyíl operátort*, amellyel a matematikában szokásoshoz hasonló szintakszissal adhatunk meg függvényeket:

function_name := *parameter(s)* -> *kifejezés*

Az Olvasó könnyen kísértésbe eshet, hogy a következő módon definiáljon függvényeket:

```
> T(t) := T0 * exp(-c*t);
```

$$\mathrm{T}(t) := T0\, e^{(-c\,t)}$$

A Maple elfogadja az utasítást. Ez biztatónak tűnik, de amint az alábbi példából kiderül, valójában nem azt csinálja, amit szerintünk jelentene a parancs.

```
> T(t), T(1/c), T(0);
```

$$T0\, e^{(-c\,t)},\ \mathrm{T}(\frac{1}{c}),\ \mathrm{T}(0)$$

Mi történt? Függvényt hoztunk létre, de nem adtuk meg a definícióját.

```
> print(T);
```

proc( ) option *remember*; 'procname(args)' end

Helyette a T eljárás *emlékezőtáblájába* írtuk be a függvény t helyen fölvett értékét. Ha az $1/c$-hez tartozó függvényértéket kérjük, ezt a Maple nem találja sem az emlékezőtáblában, sem a függvény leírásában. Így a rendszer nem is tud annál többet tenni, mint hogy kiírja a függvényhívást, gondolván, hogy a szekció folytatásában majd bizonyára hozzárendelünk T-hez egy függvényleírást.

Már találkoztunk az egyváltozós függvények esetével. Többváltozós függvények hasonlóan definiálhatók:

```
> f := (x,y) -> x^3 - 3*x*y^2;
```

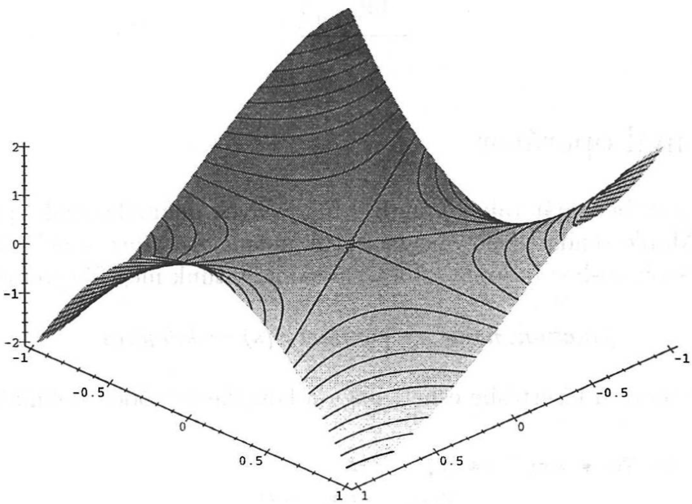
$$f := (x,\, y) \to x^3 - 3\,x\,y^2$$

```
> f(3,2);
```

$$-9$$

```
> plot3d( f, -1..1, -1..1,
> numpoints=2500, style=patchcontour, axes=framed );
```

Az így kapott grafikon a 8.4. ábrán látható. Ügyeljünk a függvénydefinícióban



8.4. ábra: Az $x^3 - 3xy^2$ függvény felületi rajza a $[0,1] \times [0,1]$ fölött

a paraméterek köré írt zárójelekre. Ezek elhagyása esetén a Maple az utasítást a következő módon interpretálja

```
> f := x, ( y -> x^3 - 3*x*y^2);
```

$$f := x,\, y \to x^3 - 3\,x\,y^2$$

vagyis az `f` nevű kifejezéssorozatot hoztuk létre, amely az `x` változóból és az $y \longmapsto x^3 - 3xy^2$ névtelen függvényből áll. Paraméter nélküli nullváltozós függvények is megadhatók.

```
> goldenratio := () -> (1+sqrt(5))/2; # constant function
```

$$goldenratio := () \to \frac{1}{2} + \frac{1}{2}\sqrt{5}$$

```
> goldenratio();
```

$$\frac{1}{2} + \frac{1}{2}\sqrt{5}$$

```
> goldenratio( x ); # no check on number of arguments
```

$$\frac{1}{2} + \frac{1}{2}\sqrt{5}$$

Legyünk óvatosak a függvénydefinícióban paraméterként nem szereplő változók használatával. Mellékhatások léphetnek föl. Néha a Maple figyelmeztet, hogy a kifejezésben szereplő változót lokálisnak deklarálja:

```
> duplicate := (x,n) -> seq(x, j = 1..n);

Warning, 'j' in call to 'seq' is not local
```

$$duplicate := (x, n) \to \mathrm{seq}(x, j = 1..n)$$

Ez egy „hamis riasztás” volt, amit az **seq** operátor implementálásának módja okozott. Máskor meg nem szól a Maple, holott valamely globális változóval nemkívánatos interferencia lépett föl.

```
> ERF := x -> 2/Pi^(1/2) * int( exp(-t^2), t = 0..x );
```

$$ERF := x \to 2\,\frac{\int_0^x e^{(-t^2)}\,dt}{\sqrt{\pi}}$$

```
> ERF(1); # indeed, ERF is nothing but the error function
```

$$\mathrm{erf}(1)$$

```
> t := 0:  # let t be assigned a value!
> ERF(1); # all goes wrong

Error, (in int) wrong number (or type) of arguments
```

Ez a probléma csak úgy oldható meg, ha t-t lokális változónak deklaráljuk. Mivel a nyíl operátoros definícióban csak egy kifejezés, egy föltételes utasítás vagy egy eljárásdefiníció szerepelhet, az eljárásokat definiáló hosszú alakot kell használnunk.

```
> ERF := proc(x)
> options operator, arrow;
> local t;
> 2/Pi^(1/2) * int( exp(-t^2), t = 0..x );
> end;
```

$$ERF := x \to \mathrm{local}\ t;\ 2\,\frac{\int e^{(-t^2)}\,dt = 0..x}{\sqrt{\pi}}$$

Ezzel megszűntek a mellékhatások.

```
> ERF(1); # indeed, ERF is nothing but the error function
```

$$\mathrm{erf}(1)$$

A 8.4. alfejezetben közelebbről is megvizsgáljuk az eljárások definiálásának általános formáját.

8.3. Szakaszonként definiált függvények

Szakaszonként definiált függvényeket is könnyen bevezethetünk a nyíl operátor segítségével. Ennek két módja van, a második a jobb.

- **A case utasításhoz hasonlóan**
  Használjunk föltételes utasítást az egyes esetek megkülönböztetésére. Az általános alak a következő:

```
parameter(s) ->
              if  1st condition then
                  1st value
              elif  2nd condition then
                  2nd value
                  ......
              elif  Nth condition then
                  Nth value
              else
                  default value
              fi
```

- **A piecewise eljárással**
  Ha a Maple rendszerben a **piecewise** eljárást alkalmazzuk, a szakaszonként definiált függvényekre vonatkozó matematikai ismeretek is elérhetők. Az általános alak a következő:

```
parameter(s) -> piecewise (
                     1st condition, 1st value
                     2nd condition, 2nd value
                     ......
                     Nth condition, Nth value
                     default value
                     )
```

Megvizsgáljuk a szakaszonként definiált függvények definiálásának mindkét módját. Először tekintsük azt a lépcsős függvényt, melynek értéke 1-nél kisebb valós számokra −1, az 1 helyen 0 és 1 különben.

```
> step := x -> if x<1 then -1 elif x=1 then 0 else 1 fi;
```

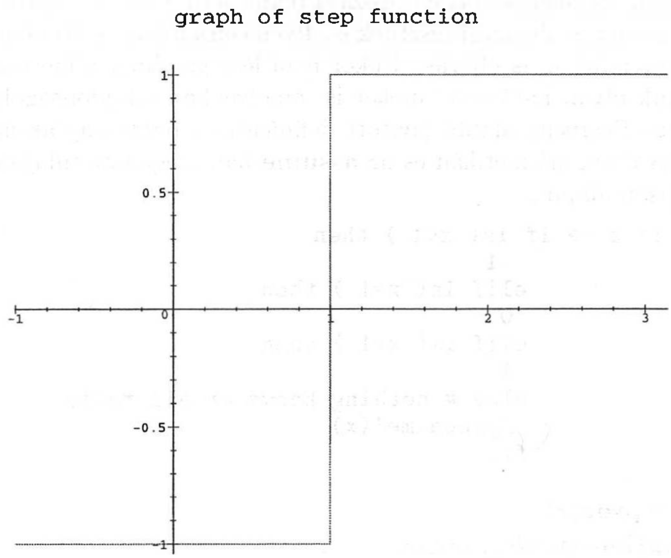
step := **proc**(x)
option*operator*, *arrow*;
if $x<1$ **then** -1 **elif** $x=1$ **then** 0 **else** 1 **fi**
end

```
> step(3/2), step(1), step(1/2);
```

$$1, 0, -1$$

```
> plot( step, -1..Pi,
> title=‘graph of step function‘, style=line );
```

A függvény Maple által kirajzolt grafikonja a 8.5. ábrán látható.



8.5. ábra: Lépcsős függvény grafikonja

Ennek a definíciónak néhány első pillantásra nem látható gyenge pontját világítja meg a következő utasítás:

```
> printlevel := 2:  # make Maple more communicative
> step(Pi);

Error, (in step) cannot evaluate boolean
 executing statement:
if x < 1 then ‘...‘ elif x = 1 then ‘...‘ else ‘...‘ fi
 step called with arguments: Pi
```

A problémát az okozza, hogy a Maple nem tudja kiértékelni az

```
if  1 < Pi  then ...
```

föltételt. A rendszer ugyanis csak a *numeric* típushoz tartozó (egész, racionális és lebegőpontos) számokat tud összehasonlítani. Mi természetesen tudjuk, hogy `1 < Pi`. Mivel a Maple képtelen logikai környezetben kiértékelni a kifejezést, a kirajzoló utasításban is a függvény nevét adtuk meg, nem pedig ezt

```
> plot( step(x), x = -1 ..  3.14 );
```

```
Error, (in step) cannot evaluate boolean
 executing statement:
if x < 1 then `...` elif x = 1 then `...` else `...` fi
 step called with arguments: x
```

Egy valós szám előjelének meghatározása tekinthető a $(0, \infty)$ nyitott intervallumra vonatkozó tartalmazási tesztnek is. Ilyen célra fölhasználható az **assume** parancs, és hívható az **is** eljárás. Ekkor nem lesz gondunk a kiértékeléssel, és használhatunk olyan határozatlanokat is, amelyekhez tulajdonságokat rendeltünk. A `step` függvény alábbi javított definíciója a pontosság aktuális értéke szerinti intervallum-aritmetikán és az **assume**-ban megadott tulajdonságokkal való számoláson alapul.

```
>   step := x -> if is( x<1 ) then
>                    -1
>                   elif is( x=1 ) then
>                    0
>                   elif is( x>1 ) then
>                    1
>                   else # nothing known or all fails
>                    'procname'(x)
>                   fi;
```

$$
\begin{aligned}
&\mathit{step} := \mathbf{proc}(x)\\
&\quad \mathbf{option}\ \mathit{operator}, \mathit{arrow};\\
&\qquad \mathbf{if}\ \mathrm{is}(x < 1)\ \mathbf{then}\ -1\\
&\qquad \mathbf{elif}\ \mathrm{is}(x = 1)\ \mathbf{then}\ 0\\
&\qquad \mathbf{elif}\ \mathrm{is}(1 < x)\ \mathbf{then}\ 1\\
&\qquad \mathbf{else}\ \text{'procname'}(x)\\
&\qquad \mathbf{fi}\\
&\quad \mathbf{end}
\end{aligned}
$$

Az eljáráson belül a `procname` speciális névvel hívatkozunk arra a névre, amivel az eljárást hívtuk (ez most `step`). Lássuk, hogyan működik a függvény:

```
>   step(Pi), step( 1 + sqrt(2)*sqrt(3) - sqrt(6) );
```

$$1, 0$$

```
>   step( exp(Pi/3*sqrt(163)) - 640319 );
```

$$1$$

```
>   assume( s > sqrt(2) ):
>   step(s);
```

$$1$$

```
>   step(u);
```

$$\mathrm{step}(u)$$

A case utasításban a logikai tesztek balról jobbra értékelődnek ki, s ha valamelyik föltétel teljesül, ez lesz a visszaadott érték, nem történik további kiértékelés. Emiatt a következő függvény definiálása és a 0 argumentumra való meghívása teljesen korrekt dolog:

```
> f := x -> if x<=0 then x else 1/x fi:
> f(0);
```

$$0$$

A föltételes utasításokkal megadott szakaszonként definiált függvények nagy hátránya, hogy matematikai műveletek nemigen végezhetők velük. Semmi ilyesmi nem lehetséges:

```
> diff( step(u), u );
```

$$\frac{\partial}{\partial u}\,\mathrm{step}(u)$$

```
> solve( step(u)=0, u );
```

$$\mathrm{RootOf}(\mathrm{step}(_Z))$$

```
> diff( g(u), u ) = step(u);
```

$$\frac{\partial}{\partial u}\,\mathrm{g}(u) = \mathrm{step}(u)$$

```
> dsolve( ", g(u) );
```

$$\mathrm{g}(u) = \int \mathrm{step}(u)\,du + _C1$$

A föltételes utasításokkal megadott szakaszonként definiált függvények sok hátránya kiküszöbölhető a **piecewise** Maple eljárással. Először tekintsük ismét az előző példát:

```
> STEP := x -> piecewise( x<1, -1,
>                         x=1, 0,
>                         x>1, 1,
>                         'procname'(x) # default
>                       );
```

$$STEP := x \to \mathrm{piecewise}(x < 1, -1, x = 1, 0, 1 < x, 1, \text{'procname'}(x))$$

```
> STEP(3/2), STEP(1), STEP(1/2), STEP(Pi);
```

$$1, 0, -1, 1$$

Nagyszerűen működik. Használhatunk változókat is:

```
> STEP(u);
```

$$\begin{cases} -1 & u < 1 \\ 0 & u = 1 \\ 1 & 1 < u \\ \mathrm{STEP}(u) & \text{otherwise} \end{cases}$$

Eredményül a függvényhívásnak megfelelő leírást kaptunk, ami további parancsokkal földolgoztatható:

```
> dsolve( diff( g(u), u ) = STEP(u), g(u) );
```

$$g(u) = \begin{cases} -u + _C1 & u < 1 \\ \text{undefined} & u = 1 \\ u - 1 + _C1 & 1 < u \end{cases}$$

A **piecewise** eljárás is sorbarendezi a logikai teszteket. A **piecewise** hívásakor azonban minden argumentum kiértékelődik és behelyettesítődik az eljárástörzsbe, még mielőtt eldőlne, hogy melyik ágon folytatódik a számítás. Ezért nem alkalmazható a következő függvénydefiníció az $x = 0$ argumentumra:

```
> f := x -> piecewise( x<=0, x, x>0, 1/x );
```

$$f := x \to \text{piecewise}(x \le 0,\, x,\, 0 < x,\, \frac{1}{x})$$

```
> f(0);

Error, (in f) division by zero
 executing statement: piecewise(x <= 0,x,0 < x,1/x)
 f called with arguments: 0
```

A **piecewise** előnye az, hogy a Maple ezekre az objektumokra alkalmazni tudja beépített matematikai ismereteit. Az ilyen szakaszonként definiált függvényeket differenciálhatjuk és integrálhatjuk, megoldhatunk őket tartalmazó (differenciál)egyenleteket, kiszámíthatjuk Taylor-sorukat stb. Ráadásul több jólismert matematikai függvény, mint például az abszolút érték, a maximum, a minimum és a Heaviside-féle lépcsősfüggvény a **piecewise**-nak megfelelő formára konvertálható a további számítások elvégzéséhez. Itt elég lesz egyetlen példa; továbbiakkal majd a diferenciálásról és az integrálásról, a sorfejtésekről, a határértékekről és a differenciálegyenletek megoldásáról szóló fejezetekben találkozhat az Olvasó.

```
> f := min( x - 1, x^2 - 4 );
```

$$f := \min(x - 1,\, x^2 - 4)$$

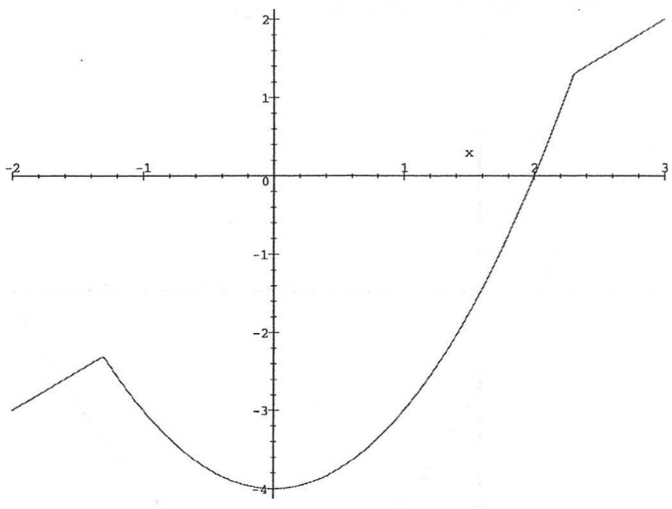
```
> plot( f, x = -2..3 );
```

A megfelelő grafikont a 8.6. ábra tartalmazza.

Az f függvényt így konvertáltathatjuk a szakaszonkénti definíciónak megfelelő alakra:

```
> F := convert( f, piecewise );
```

$$F := \begin{cases} x - 1 & x \le \frac{1}{2} - \frac{1}{2}\sqrt{13} \\ x^2 - 4 & x \le \frac{1}{2} + \frac{1}{2}\sqrt{13} \\ x - 1 & \frac{1}{2} + \frac{1}{2}\sqrt{13} < x \end{cases}$$



8.6. ábra: A $\min(x-1, x^2-4)$ függvény grafikonja a $(-2,3)$ intervallumon

Ezután már differenciálhatjuk F-et x szerint, megoldhatunk F-et tartalmazó egyenleteket vagy kiszámíthatjuk valamely pont körüli Taylor-sorát.

```
> diff( F, x );
```

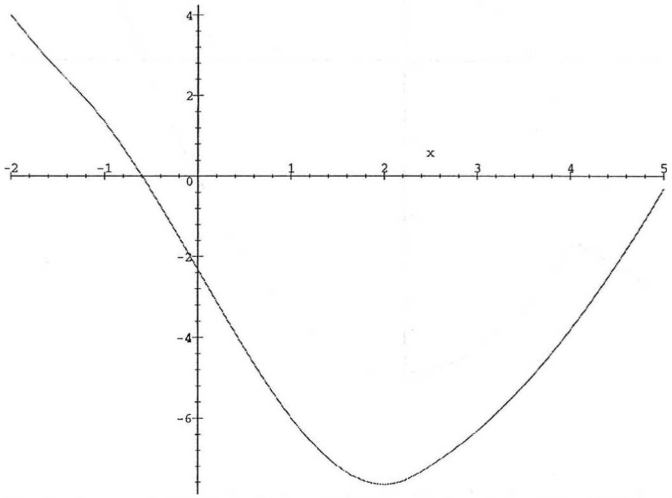
$$\begin{cases} 1 & x < \frac{1}{2} - \frac{1}{2}\sqrt{13} \\ \mathit{undefined} & x = \frac{1}{2} - \frac{1}{2}\sqrt{13} \\ 2\,x & x < \frac{1}{2} + \frac{1}{2}\sqrt{13} \\ \mathit{undefined} & x = \frac{1}{2} + \frac{1}{2}\sqrt{13} \\ 1 & \frac{1}{2} + \frac{1}{2}\sqrt{13} < x \end{cases}$$

```
> integrate( F, x );
```

$$\begin{cases} -x + \frac{1}{2}x^2 & x \leq \frac{1}{2} - \frac{1}{2}\sqrt{13} \\ -\frac{13}{12}\sqrt{13} - 4\,x + \frac{19}{12} + \frac{1}{3}x^3 & x \leq \frac{1}{2} + \frac{1}{2}\sqrt{13} \\ \frac{1}{2}x^2 - \frac{13}{6}\sqrt{13} - x & \frac{1}{2} + \frac{1}{2}\sqrt{13} < x \end{cases}$$

```
> plot( ", x = -2..5 );
```

A grafikon a 8.7. ábrán látható.



8.7. ábra: $\int \min(x-1, x^2-4)\,dx$ grafikonja a $(-2, 5)$ intervallumon

```
> solve( F=-3, x );
```

$$-2, 1, -1$$

```
> series( F, x=1, 3 );
```

$$-3 + 2\,(x-1) + (x-1)^2$$

```
> convert( ", polynom );
```

$$-5 + 2\,x + (x-1)^2$$

```
> expand( " );
```

$$x^2 - 4$$

8.4. Maple eljárások

Vessünk még egy pillantást az előző alfejezetben a nyíl operátorral definiált `step` függvényre: az ott látható output egy *eljárás*. A Maple eljárások definíciója általában a következő szintakszist követi:

proc(*parameter_sequence*)
  [**local** *name_sequence*;]
  [**global** *name_sequence*;]
  [**options** *name_sequence*;]
  statements
end

ahol a *name_sequence* Maple nevek vesszővel elválasztott sorozata. Sok esetben a *parameter_sequence* ugyancsak Maple nevek sorozata, ám ha a dinamikus típusellenőrzést használjuk, akkor a nevek után a `::` dupla kettősponttal elválasztva típusukat is megadhatjuk.

```
> sgn := proc( n::integer ) (-1)^n end:
> sgn(Pi);

Error, sgn expects its 1st argument, n, to be of type
integer, but received Pi

> sgn(-2);
```

1

Az **sgn** eljárásnak egy paramétere van. Az eljárás meghívásakor a Maple ellenőrzi, hogy a megadott argumentum *integer* típusú-e. Ha nem, vagy ha egyáltalán nem adtunk meg argumentumot, akkor a rendszer hibaüzenetet küld, egyébként pedig kiszámolja a hatvány értékét.

Általában az eljárás által visszaadott érték az utoljára végrehajtott utasítás értéke (hacsak nem használjuk explicit módon a **RETURN** utasítást). Nézzünk egy példát:

```
> MEMBER := proc( x::anything, L::list, pos::name )
>   local i;
>   for i to nops(L) do
>     if L[i] = x then pos := i; RETURN(true) fi
>   od:
>   false
> end:
> MEMBER( 2, [2,1,3,2], 'position' );
```

true

```
> position;
```

1

```
> MEMBER( 4, [2,1,3,2] );
```

false

Ha egy eljárás definíciójában egy olyan változónak adunk értéket, amelyet nem definiáltunk sem lokálisként, sem globálisként, akkor a Maple automatikusan lokálisnak föltételezi. A következő példa ezt illusztrálja:

```
> f := proc(x)
> y := 1 + sin(x);
> y^2 + y + 1/y + 1/y^2
> end;
```

```
Warning, `y` is implicitly declared local
```

$$f := \mathbf{proc}(x)\,\mathbf{local}\,y;\ y := 1 + \sin(x)\ ;\ y^2 + y + 1/y + 1/y^2\ \mathbf{end}$$

```
> y := unspecified:  f(Pi);
```

$$4$$

Magától értetődik, hogy jobb megadni a `local` és a `global` deklarációkkal az eljárásdefinícióban használt változók hatókörét. Ezek a deklarációk bármilyen sorrendben következhetnek, de mindegyik legföljebb egyszer fordulhat elő.

A Maple-ben lehetőség van úgynevezett *indexelt függvényhívásra.* A rendszer ezt használja például a különböző alapú logaritmus függvények hívásai során. **log**[**b**] a b alapú logaritmust hívja. Itt csak egy példát mutatunk. Ez talán elegendő ahhoz, hogy némi elképzelésünk legyen az indexelt függvények definíciójáról. Kiterjesztjük a Maple Euler-féle béta függvény definíciójának implementációját a következő módon definiált nem teljes béta függvényre

$$B_x(p,q) = \int_0^x t^{p-1}(1-t)^{q-1}\,dt.$$

$B_1(p,q)$ tehát a szokásos (teljes) béta függvény. A Maple-ben a $B_x(p,q)$ egy kezdetleges, de nem „bolondbiztos” implementációja lehet a következő:

```
> BETA := proc(p,q)
>   local x:
>   if type( procname, indexed ) then
>     x := op( procname );
>     int( t^(p-1)*(1-t)^(q-1), t=0..x )
>   else
>     Beta(p,q)
>   fi
>   end:
> BETA[1](2,3), Beta(2,3);
```

$$\frac{1}{12}, \frac{1}{12}$$

```
> BETA(3/2,5/2);
```

$$\frac{1}{16}\pi$$

A **BETA** eljárás hívásakor a Maple először a tényleges eljárásnevet vizsgálja meg alaposabban. Ellenőrzi, hogy az eljárásban belül szereplő `procname` változó értéke **BETA**[x] alakú indexelt név-e. Ha nem, föltételezi, hogy **BETA** a szokásos (teljes) béta függvényt jelenti. Ha indexelt nevet használtunk, az x lokális változó az index értékét veszi föl, és a nem teljes béta függvénynek megfelelő integrál kerül meghatározásra.

8.5. Rekurzív eljárások

A Maple-ben definiálhatunk rekurzív függvényeket és eljárásokat. Ezt a Lucas féle L_n számok meghatározásával illusztráljuk, melyeket a következő lineáris rekurzió definiál

$$L_1 = 1, L_2 = 3 \text{ és } L_n = L_{n-1} + L_{n-2} \text{ ha } n > 2.$$

A képlet a nyíl operátor segítségével egyből átírható a következő formába:

```
> L := n ->
>   if not type( n, posint )
>   then ERROR( `wrong type of arguments` )
>   elif n=1 then 1
>   elif n=2 then 3
>   else L(n-1) + L(n-2)
>   fi;
```

```
L := proc(n)
  option operator, arrow;
    if not type(n, posint) then ERROR(`wrong type of arguments`)
    elif n = 1 then 1
    elif n = 2 then 3
    else L(n - 1) + L(n - 2)
    fi
  end
```

Az **ERROR** eljárást a függvény végrehajtása során hibaüzenet generálására használtuk, ha a megadott argumentum nem pozitív egész szám volt. Kicsit kényelmesebb a Maple eljárások definiálásának következő szabványos módja:

```
> L := proc( n::posint )
>   if n = 1 then 1
>   elif n = 2 then 3
>   else L(n-1) + L(n-2)
>   fi
>   end:
> L(6);
```

$$18$$

Ez azonban nem valami hatékony módja a Lucas-féle számok kiszámításának. Számoltassuk meg a Maple eljáráshívások számát az **exprofile** eljárással (részletesebb leírása a help rendszerben található meg).

```
> readlib( exprofile ):  # load the library function
> kernelopts( profile = true ):  # enable profiling
> writeto( 'output' ); # redirect output to file
```

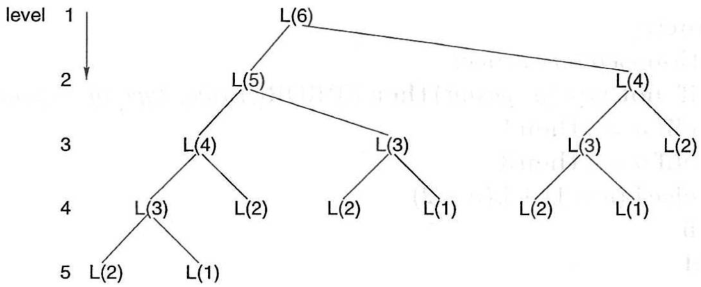
Amíg ebben a módban vagyunk, a prompt jel nem jelenik meg a képernyőn.

```
> L(6):
> kernelopts( profile = false ):  # disable profiling
> writeto( 'terminal' ); # redirect output to screen
```

```
> exprofile( 'output' );

name                    #calls
====                    ======
Main_Routine                 1
type/posint                 15
L                           15
```

A táblázatból elhagytuk a számítási időre és a memóriahasználatra vonatkozó adatokat. A fönti információk rögtön világossá válnak, ha megértjük, hogyan számítja ki a Maple a Lucas-számokat. `L(6)` meghatározásához ki kell számítani `L(5)`-öt és `L(4)`-et. Ezek mindegyikéhez két újabb függvényhívásra van szükség, s ez így megy mindaddig, míg az összes szükséges Lucas-számot meg nem kaptuk. A 8.8. ábra tartalmazza az `L(6)` meghatározása közben végrehajtott függvényhívások gráfját. A gráf, az **exprofile** eredményével összhangban,



8.8. ábra: $L(6)$ rekurzív kiszámítása a `remember` opció használata nélkül

világosan mutatja, hogy `L(2)`-t például ötször is kiszámoltuk. Az **L** eljárás fönti definíciója mellett tehát exponeciális (2^n) idő szükséges az n-dik Lucas-szám kiszámításához, ezért `L(100)` meghatározása a végtelenségig elhúzódna. De ennél okosabban is eljárhatunk. Nyilvánvaló, hogy „emlékeznünk" kellene az egyszer már kiszámolt függvényértékekre, hogy szükség esetén újra föl tudjuk használni őket. Ezt a Maple `remember` opciója segítségével érhetjük el.

```
> L := proc( n::posint ) Lucas( n ) end:
> Lucas := proc(n)
>   option remember;
>   if n = 1 then 1
>   elif n = 2 then 3
>   else Lucas(n-1) + Lucas(n-2)
> fi
> end:
> kernelopts( profile = true ):
> writeto( 'output' ):
> L(6):
> kernelopts( profile = false ): # disable profiling
> writeto( 'terminal' ); # redirect output to screen
```

```
> exprofile( 'output' );

name                      #calls
====                      ======
Main_Routine                   1
Lucas                          6
type/posint                    1
L                              1
```

Az **L** új definíciójával az időkomplexitás lineáris ($2n$). **L** hívásakor a következő történik: az argumentum típusának ellenőrzése után a **Lucas** rekurzív eljárás kerül meghívásra. Így az argumentum típusát a számítás során csak egyszer ellenőrizzük (és nem újra meg újra minden egyes rekurzív híváskor). Minden Maple eljárásnak van egy *emlékezőtáblája*. Ez a tábla sokszor nem létezik abban az értelemben, hogy a rá mutató pointer értéke `NULL`. De ha az eljárás definíciójában megadjuk az `option remember` utasítást, a Maple emlékezőmechanizmusa azonnal aktiválódik. Az emlékezőtábla elemei a híváskor megadott argumentumokkal indexelt függvényértékek lesznek. Amikor a példánkban szereplő **Lucas** eljárást valamely n argumentummal meghívjuk, a Maple először az emlékezőtáblában nézi meg, hogy a `Lucas(n)` értéket korábban kiszámolta-e már. Ha igen, akkor a táblabeli értéket adja vissza. Egyébként pedig végrehajtja az eljárás törzsében definiált utasításokat, és automatikusan elhelyezi az `(n, Lucas(n))` értékpárost a **Lucas** emlékezőtáblájában. Így minden Lucas számot csak egyszer számítunk ki.

Az előző függvényhívási gráfnak megfelelő terminológiával kifejezve módszerünk a következőt jelenti: „járjuk be a gráfot a mélység szerinti (depth first) bejárási stratégiával, minden függvényértéket csak egyszer számítsunk ki, a kapott eredményt tároljuk, és szükség esetén ezt a tárolt értéket használjuk föl újra".

Az eljárások definíciójával együtt emlékezőtáblájukat is kiírathatjuk, ha a `verboseproc` interfész változó értékét 3-ra állítjuk. Esetünkben ez a következőt eredményezi:

```
> interface( verboseproc = 3 ):
> print( Lucas );
```

proc(n)

option*remember*;

if $n = 1$ **then** 1 **elif** $n = 2$ **then** 3 **else** $\text{Lucas}(n-1) + \text{Lucas}(n-2)$ **fi**

end

$\# (5) = 11$

$\# (4) = 7$

$\# (2) = 3$

$\# (1) = 1$

$\# (6) = 18$

$\# (3) = 4$

Az eljárás emlékezőtáblája az eljáráshoz rendelt struktúra negyedik operanduszaként érhető el.

```
> op( 4, eval(Lucas) ); # remember table of Lucas
```

table([
$4 = 7$
$5 = 11$
$6 = 18$
$1 = 1$
$2 = 3$
$3 = 4$
])

Az emlékezőtábla közvetlenül is módosítható *procedure(argument) := value* alakú *funkcionális értékadásokkal*. Ezt mutatjuk be a továbbiakban. Először is kezdjünk új Maple szekciót:

```
> restart:
> Lucas := proc(n) Lucas(n) := Lucas(n-1) + Lucas(n-2) end:
> Lucas(1) := 1: Lucas(2) := 3:
> op( 4, eval(Lucas) ); # initial remember table
```

table([
$1 = 1$
$2 = 3$
])

```
> Lucas(4): op( 4, eval(Lucas) ); # updated remember table
```

table([
$1 = 1$
$2 = 3$
$3 = 4$
$4 = 7$
])

A Maple **forget** segédfüggvényével törölhetjük az emlékezőtábla egy vagy több elemét:

```
> readlib( forget ): # load the utility function
> forget( Lucas, 3): # forget the result of Lucas(3);
> op( 4, eval(Lucas) ); # updated remember table
```

table([
$1 = 1$
$2 = 3$
$4 = 7$
])

Ha a **Lucas** mellett minden olyan eljárás emlékezőtáblájának elemeit is törölni akarjuk, amelynek neve a `Lucas/` karakterekkel kezdődik, akkor a következőt írhatjuk:

```
> forget( Lucas ):
> op( 4, eval(Lucas) ); # clear remember table
```

table([
])

A **Lucas** és a `Lucas/` karakterekkel kezdődő nevű eljárások emlékezőtáblájának eltávolítása :

```
> Lucas := subsop( 4 = NULL, eval(Lucas) ):
> op( 4, eval(Lucas) ); # NULL pointer to remember table
>
```

Az Olvasó megkérdezhetné, hogy a **forget**(<*name*>) parancs a *name* nevű eljárás mellett vajon miért törli az összes olyan eljárás emlékezőtábláját is, amelynek neve a *name/* karakterekkel kezdődik. Ennek oka az, hogy ily módon az **int**, az **evalc** vagy a hozzájuk hasonló rutinok tábláit anélkül törölhetjük, hogy ismernünk kellene a táblákat használó összes **int**/... vagy **evalc**/... nevű segédfüggvényt. Tehát a **forget** lehetővé teszi, hogy tiszta lappal induljunk. A **forget** viselkedésének további részleteit megismerhetjük a beépített help rendszerből.

8.6. Az **unapply** függvény

Új függvények létrehozásának létezik egy harmadik módja is; ez különösen akkor kényelmes, ha egy Maple szekció közepén jutunk el egy olyan szimbolikus kifejezéshez, amelyet egy függvény megadására szeretnénk fölhasználni. Ezt az **unapply** eljárással tehetjük meg. Ahogy a nevéből is kikövetkeztethető, ez épp az ellenkezője annak, amikor egy függvényt alkalmazunk szimbolikus paraméterekre.

```
> formula := ( b^2*x^2*sin(b*x) - 2*sin(b*x)
> + 2*b*x*cos(b*x)*a*t ) / b^3;
```

$$formula := \frac{b^2\,x^2\sin(b\,x) - 2\sin(b\,x) + 2\,b\,x\cos(b\,x)\,a\,t}{b^3}$$

```
> F := unapply( formula, x, t );
```

$$F := (x, t) \to \frac{b^2\,x^2\sin(b\,x) - 2\sin(b\,x) + 2\,b\,x\cos(b\,x)\,a\,t}{b^3}$$

```
> F(0,1), F(Pi/b,5);
```

$$0, -10\,\frac{\pi\,a}{b^3}$$

Ha a nyíl operátorral megadott függvénydefinícióban a " operátorral próbálunk hivatkozni az előző kifejezésre, kudarcot vallunk:

```
> formula := ( b^2*x^2*sin(b*x) -2*sin(b*x)
> + 2*b*x*cos(b*x)*a*t ) / b^3:
> F := (x,t) -> ";
```

$$F := (x, t) \to "$$

```
> F(u,v);
>
```

A problémát az okozza, hogy a " operátor valójában egy *környezeti változó*, amely az **F** eljáráson belül lokálisan működik, de nem hivatkozhatunk vele semmiféle ezen kívüli objektumra. Az eljáráshívás előtti értéke a verembe került, és csak az eljárásból való visszatérés után állítódik vissza:

```
> G := (x,t) -> formula;
```

$$G := (x, t) \to formula$$

```
> G(1,2);
```

$$\frac{b^2\,x^2\sin(b\,x) - 2\sin(b\,x) + 2\,b\,x\cos(b\,x)\,a\,t}{b^3}$$

Most pontosan azt kaptuk, amit kértünk: bármilyen argumentumokkal hívjuk is **G**-t, a `formula` kiértékeléséből származó értéket kapjuk vissza. A függvény leírását tehát valójában *paraméter* -> *description* formában kell begépelnünk. Az egyetlen alternatíva az lehet, hogy a kifejezést egy globális változó, mondjuk a body segítségével helyettesítjük be a függvény leírásába:

```
> H := subs( body = formula, (x,t) -> body );
```

$$H := (x, t) \to \frac{b^2\,x^2\sin(b\,x) - 2\sin(b\,x) + 2\,b\,x\cos(b\,x)\,a\,t}{b^3}$$

```
> H(u,v);
```

$$\frac{b^2\,u^2\sin(b\,u) - 2\sin(b\,u) + 2\,b\,u\cos(b\,u)\,a\,v}{b^3}$$

8.7. Függvényműveletek

A függvényekkel végzett elemi műveletek, mint például az összeadás, a szorzás vagy a kompozíció, könnyen végrehajthatók a Maple rendszerben. Néhány példa:

```
> f := x -> ln(x) + 1:  g := y -> exp(y) - 1:
> h := f + g:  h(z);
```

$$\ln(z) + e^z$$

```
> h := f * g:  h(z);
```

$$(\ln(z) + 1)\,(e^z - 1)$$

```
> h := f @ g:  h(z);
```

$$\ln(e^z - 1) + 1$$

```
> h := g @ f:  h(z);
```

$$e^{(\ln(z)+1)} - 1$$

```
> simplify(");
```

$$z\,e - 1$$

```
> (f@@4)(z); # equivalent to f(f(f(f(z))))
```

$$\ln(\ln(\ln(\ln(z) + 1) + 1) + 1) + 1$$

A @ operátort a **macro** paranccsal kombinálva a Maple hatékony mechanizmust nyújt rövidítések bevezetésére. Ennek egyik példája a helyettesítés. Emlékezzünk vissza, hogy a 6.3. alfejezet szerint a **subs** csak a helyettesítést végzi el, de a végén nem végez kiértékelést.

```
> subs( n=2, Zeta(n) ); # substitution
```

$$\zeta(2)$$

```
> "; # evaluation
```

$$\frac{1}{6}\pi^2$$

Tegyük fel, hogy a helyettesítés végeredményét mindig ki akarjuk értékelni, ekkor a következőt tehetjük:

```
> macro( subs = eval @ subs ): # new version of subs
> subs( n=2, Zeta(n) ); # with new version of subs
```

$$\frac{1}{6}\pi^2$$

8.8. Névtelen függvények

Nem vagyunk kötelesek nevet adni a függvényeknek. A *névtelen függvények* hasznosak lehetnek, ha például valamely műveletet csak egyszer akarunk végrehajtani, és nem akarjuk időnket azzal vesztegetni, hogy ennek még külön nevet is adjunk. Névtelen függvényeket leginkább a **map**, **select**, **remove**, **zip** és a **collect** eljárásokkal kapcsolatban használunk. Néhány példa:

```
> map( x -> x + 2, [1,2,3] ); # add 2 to list elements
```

$$[3, 4, 5]$$

```
> map( x -> x^2, a + b + c ); # square summands
```

$$a^2 + b^2 + c^2$$

```
> data := [[1,1.0], [2,3.8], [3,5.1] ]:
> # take the logarithm of the 2nd element of each entry
> map( x-> applyop(ln,2,x), data );
```

$$[[1, 0], [2, 1.335001067], [3, 1.629240540]]$$

```
> # compute sums of derivative and antiderivative
> map( (f,x) -> diff(f,x) + int(f,x),
> [cos(z), sin(z), exp(z), ln(z) ], z );
```

$$[0, 0, 2\,e^z, \frac{1}{z} + z\ln(z) - z]$$

```
> sum( 'x^i', 'i' = 0..6 );
```

$$1 + x + x^2 + x^3 + x^4 + x^5 + x^6$$

```
> # select low and high terms in polynomial
> select( t -> degree(t)<3, " );
```

$$1+x+x^2$$

```
> remove( t -> degree(t)<3, "" );
```

$$x^3+x^4+x^5+x^6$$

```
> # combine two lists of data
> zip( (x,y) -> [x,ln(y)], [1,2,3], [1.0,3.8,5.1] );
```

$$[[1, 0], [2, 1.335001067], [3, 1.629240540]]$$

```
> # collect a polynomial in x and square coefficients
> collect( x^2 + y*x^2 + 2*x +y, x, z -> z^2);
```

$$(1+y)^2\, x^2 + 4\,x + y^2$$

8.9. Gyakorlatok

1. Oldjuk meg az $x^3-(a-1)x^2+a^2x-a^3=0$ egyenletet x-re. Határozzuk meg azt a függvényt, amely az előző egyenlet első megoldását adja a függvényében, és számoljuk ki a megoldást $a=0$ és $a=1$ esetén. Adjunk közelítő megoldást $a=2$-re.

2. Definiáljunk egy Maple függvényt, amelynek értéke 1 a $[-1,1]$ intervallumon és 0 különben. Rajzoltassuk is ki a függvény gráfikonját.

3. Definiáljuk a következő függvényt $f: t \longmapsto \sum_{n=1}^{8}(-1)^{n+1}\frac{2}{n}\sin(nt)$. Számítsuk ki $f(\frac{\pi}{10})$-et és $f(\frac{\pi}{6})$-ot. Rajzoltassuk ki a függvény gráfikonját.

4. Mi lesz `f(1)`, `f(4)` és `f()` értéke a következő értékadások után?

```
> x := 1:
> f := proc(x) 2 end:
> f(x) := 3:
```

5. Írjunk Maple eljárást az $L_n(x)$ Legendre-polinomok kiszámítására. Ezek a polinomok az $L_0(x)=1, L_1(x)=x$ és $L_n(x)=\frac{n-1}{n}(x\,L_{n-1}(x)-L_{n-2}(x))+x\,L_{n-1}(x)$ ha $x>1$ rekurzív relációt elégítik ki. Számítsuk ki $L_7(x)$-et, és ellenőrizzük az eredményt a Maple **orthopoly**[**P**] eljárásával. Ki tudja-e számítani a megírt eljárás $L_{50}(x)$-et?

6. Írjunk olyan névtelen függvényt, amely egészek egy halmazából kiválasztja azokat, amelyek 0 és 10 közé esnek. A **rand** eljárással generáltassunk 100 egész számból álló halmazt, és alkalmazzuk névtelen függvényünket erre a halmazra.

7. Írjunk olyan névtelen függvényt, amely egy kétváltozós polinomból (ilyet a **randpoly** eljárás segítségével is létrehozhatunk) elhagyja mindazon tagokat, amelyeknek negatív az együtthatója.

9. Differenciálás

Ez a fejezet elmagyarázza a szimbolikus differenciálásra szolgáló **diff** és **D** eljárásokat, példákat mutat az implicit differenciálásra, és röviden tárgyalja az automatikus differenciálás lehetőségeit a Maple-ben.

9.1. Szimbolikus differenciálás

A **diff** Maple eljárással formulákat differenciálhatunk:

```
> 'diff( exp(-x^2), x)';
```

$$\frac{\partial}{\partial x} e^{(-x^2)}$$

```
> ";
```

$$-2\, x\, e^{(-x^2)}$$

Az első parancs körüli aposztrófok arra szolgálnak, hogy késleltessék a derivált kiszámítását, és kétdimenziós formában jelenítsék meg az inputot. Ugyanezt elérhettük volna a **Diff** tétlen paranccsal, amely csupán a kiértékeletlen eredményt adja. Ezután explicit módon kérnünk kell a kifejezés értékét:

```
> Diff( ln(x/(x^2+1)), x ):  " = value(");
```

$$\frac{\partial}{\partial x} \ln(\frac{x}{x^2+1}) = \frac{(\frac{1}{x^2+1} - 2\,\frac{x^2}{(x^2+1)^2})\,(x^2+1)}{x}$$

```
> normal(");
```

$$\frac{\partial}{\partial x}\ln(\frac{x}{x^2+1}) = -\frac{x^2-1}{x\,(x^2+1)}$$

Egy igazi Maple szekcióban először valószínűleg a **Diff** tétlen eljárást alkalmaznánk, csupán azért, hogy lássuk, a kívánt formulát gépeltük-e be. Ezután, ha valóban a differenciálandó formulát kaptuk, a munkalapos fölhasználói felület szolgáltatásait alkalmazva kicserélhetjük a **Diff**-et az „aktív" **diff** eljárásra, vagy a **value**-val kérhetjük kiértékelését (a karakteres fölhasználói felület esetében a beépített sorszerkesztőt használhatjuk). Az alábbi példákban a könnyebb érthetőség és az eredmények jobb megvilágítása miatt a **Diff**-et fogjuk alkalmazni.

```
> Diff( x^(x^x), x ):  " = value(");
```

$$\frac{\partial}{\partial x}x^{(x^x)} = x^{(x^x)}\,(x^x\,(\ln(x)+1)\ln(x) + \frac{x^x}{x})$$

```
> collect( ", ln(x), simplify );
```

$$\frac{\partial}{\partial x}x^{(x^x)} = x^{(x^x+x)}\ln(x)^2 + x^{(x^x+x)}\ln(x) + x^{(x^x+x-1)}$$

A differenciálási szabályok egyszerűsítések nélküli merev alkalmazása nagyon gyorsan áttekinthetetlen eredményekhez vezethet. A kifejezések ilyen „fölfúvódásától" eltekintve a magasabb rendű deriváltak kiszámítása sem okoz bonyodalmakat:

```
> Diff( x^(x^x), x, x ):  " = value("):
> collect( ", ln(x), simplify );
```

$$\begin{aligned}\frac{\partial^2}{\partial x^2}x^{(x^x)} &= x^{(x^x)}\,(x^2)^x\ln(x)^4 + (x^{(x^x+x)} + 2\,x^{(x^x)}\,(x^2)^x)\ln(x)^3\\ &+ (2\,x^{(x^x+x)} + 2\,x^{(x^x-1)}\,(x^2)^x + x^{(x^x)}\,(x^2)^x)\ln(x)^2\\ &+ (3\,x^{(x^x+x-1)} + x^{(x^x+x)} + 2\,x^{(x^x-1)}\,(x^2)^x)\ln(x) + x^{(x^x-2)}\,(x^2)^x\\ &+ 2\,x^{(x^x+x-1)} - x^{(x^x+x-2)}\end{aligned}$$

```
> Diff( exp(-x^2), x$5 ):  " = value(");
```

$$\frac{\partial^5}{\partial x^5}e^{(-x^2)} = -120\,x\,e^{(-x^2)} + 160\,x^3\,e^{(-x^2)} - 32\,x^5\,e^{(-x^2)}$$

Az input lerövidítésére a $ sorozat operátort használtuk. A

```
diff( exp( -x^2), x$5 )
```

kifejezés ekvivalens a hosszabb

```
diff( exp( -x^2), x,x,x,x,x )
```

alakkal.

<u>Megjegyzés:</u> **diff**(*expression*, [*var$n*]) a magasabb rendű deriváltak alternatív jelölésmódja. Azért vették hozzá a Maple rendszerhez, hogy az üres lista megadásával hivatkozni tudjunk a 0-dik deriváltra, vagyis az eredeti kifejezésre:

```
> for i from 0 to 8 by 4 do
> ``.i.`th derivative` = diff( x^8 / 1680, [x$i] )
```

```
> od;
```

$$0th\ derivative = \frac{1}{1680}x^8$$
$$4th\ derivative = x^4$$
$$8th\ derivative = 24$$

Az x változó egyenlettel definiált y függvényének differenciálása a következő elegáns módon végezhető el.

```
> alias( y = y(x) ):  # consider y as a function of x
> eq := x^2 + y^2 = c; # equation defining y; c is constant
```

$$eq := x^2 + y^2 = c$$

```
> diff(eq,x);
```

$$2\,x + 2\,y\,(\frac{\partial}{\partial x}\,y) = 0$$

```
> dydx := solve( ", diff(y,x) ); # 1st derivative
```

$$dydx := -\frac{x}{y}$$

```
> diff(eq,x$2);
```

$$2 + 2\,(\frac{\partial}{\partial x}\,y)^2 + 2\,y\,(\frac{\partial^2}{\partial x^2}\,y) = 0$$

```
> solve( ", diff(y,x$2) ); # 2nd derivative
```

$$-\frac{1 + (\frac{\partial}{\partial x}\,y)^2}{y}$$

```
> d2ydx2 := normal( subs( diff(y,x) = dydx, " ) );
```

$$d2ydx2 := -\frac{x^2 + y^2}{y^3}$$

Az eredeti egyenletet mellékföltételnek tekintve tovább egyszerűsíthetünk:

```
> d2ydx2 := simplify( ", {eq}, [x,y,c] );
```

$$d2ydx2 := -\frac{c}{y^3}$$

```
> d3ydx3 := diff(",x);
```

$$d3ydx3 := 3\,\frac{c\,(\frac{\partial}{\partial x}\,y)}{y^4}$$

```
> d3ydx3 := normal( subs( diff(y,x) = dydx, " ) );
```

$$d3ydx3 := -3\,\frac{c\,x}{y^5}$$

A magasabb rendű deriváltak ugyanígy számíthatók ki. A fejezet végén az implicit differenciálásra szolgáló **implicitdiff** eljárást is ismertetni fogjuk.

```
> alias( y = y ): # unalias y for further usage
```

A parciális deriváltak sem okoznak gondot a **diff**-nek. Két kétváltozós függvényt tartalmazó példa:

```
> Diff(exp(a*x*y^2), x, y$2 ): " = factor( value(") );
```

$$\frac{\partial^3}{\partial y^2\,\partial x}\,e^{(a\,x\,y^2)} = 2\,a\,e^{(a\,x\,y^2)}\,(1 + 5\,a\,x\,y^2 + 2\,a^2\,y^4\,x^2)$$

```
> Diff( sin(x+y)/y^4, x$5, y$2 ): " = value(");
```

$$\frac{\partial^7}{\partial y^2\,\partial x^5}\,\frac{\sin(x+y)}{y^4} = -\frac{\cos(x+y)}{y^4} + 8\,\frac{\sin(x+y)}{y^5} + 20\,\frac{\cos(x+y)}{y^6}$$

```
> collect( ", cos(x+y), normal );
```

$$\frac{\partial^7}{\partial y^2\,\partial x^5}\,\frac{\sin(x+y)}{y^4} = -\frac{(y^2-20)\cos(x+y)}{y^6} + 8\,\frac{\sin(x+y)}{y^5}$$

Ha nem formulát, hanem függvényt szeretnénk differenciálni, akkor a **D** operátort alkalmazhatjuk. **D** függvényként hat, az általa előállított derivált is függvény lesz. Ez különösen akkor kényelmes, ha a függvény deriváltját akarjuk kiszámítani vagy specifikálni egy adott pontban.

```
> g := x -> x^n * exp(sin(x)); # the function
```

$$g := x \to x^n\,e^{\sin(x)}$$

```
> D(g); # the derivative
```

$$x \to \frac{x^n\,n\,e^{\sin(x)}}{x} + x^n\cos(x)\,e^{\sin(x)}$$

```
> D(g)(Pi/6); # the derivative at Pi/6
```

$$6\,\frac{(\frac{1}{6}\,\pi)^n\,n\,e^{(1/2)}}{\pi} + \frac{1}{2}\,(\frac{1}{6}\,\pi)^n\,\sqrt{3}\,e^{(1/2)}$$

`D(f)` lényegében ugyanaz, mint `unapply( diff(f(x),x), x )`.

Lényeges, hogy világosan lássuk a **diff** és a **D** közti különbséget: a **diff** *formulát* differenciál és formulát ad vissza, míg **D** *leképezést* differenciál és leképezést is ad eredményül. Néhány példa:

```
> diff( cos(t), t ); # derivative of a formula
```

$$-\sin(t)$$

```
> D(cos); # derivative of a function
```

$$-\sin$$

```
> (D@@2)(cos); # 2nd derivative of a function
```

$$-\cos$$

```
> D(cos)(t); # derivative of a function at some point
```

$$-\sin(t)$$

```
> D( cos(t) );
```

$$D(\cos(t))$$

Az utolsó parancsban a Maple cos(t)-t nem a koszinusz függvényként fogja föl, de nem is úgy, mint a koszinusz és valamely t függvény kompozícióját. Az utóbbi cél eléréséhez szigorúan követnünk kell a Maple szintakszisát és szemantikáját, ezért a @ kompozíció operátort kell használnunk:

```
> D( cos @ t);
```

$$(-\sin)@t\,D(t)$$

Ha azonban azt tesszük föl, hogy t konstans, akkor már valóban hajlandó a Maple a cos(t)-t függvénynek tekinteni, ti. konstans függvénynek:

```
> assume( t, constant ): D( cos(t) );
```

$$0$$

Még könnyebben érthető a dolog, ha egy ismerős konstanst használunk:

```
> D( cos(1) );
```

$$0$$

A cos(1) szám konstans függvénynek tekinthető, ezért deriváltja a 0 függvény:

```
> t := 't': # unassign t, i.e., forget assumption
> diff( cos, t );
```

$$0$$

Az utolsó példában a Maple a cos-t olyan kifejezésnek tekintette, amelyben a t nem fordul elő.

Ha implicit függvényt akarunk definiálni, a leképezést kényelmesebb kifejezésként kezelni és a **D**-vel elvégeztetni a differenciálást. Az előzőekben a **diff**-fel megoldott példa az új formalizmus szerint a következő lesz:

```
> eq := x^2 + y^2 = c: D( eq );
```

$$2\,D(x)\,x + 2\,D(y)\,y = D(c)$$

Megmondjuk, hogy x független változó, c pedig konstans:

```
> D(x) := 1: D(c) := 0: dydx := solve( "", D(y) );
```

$$dydx := -\frac{x}{y}$$

```
> (D@@2)( eq );
```

$$2 + 2\,(D^{(2)})(y)\,y + 2\,D(y)^2 = 0$$

```
> solve( ", (D@@2)(y) );
```

$$-\frac{1 + D(y)^2}{y}$$

```
> d2ydx2 := normal( subs( D(y)=dydx, " ) );
```

$$d2ydx2 := -\frac{x^2 + y^2}{y^3}$$

És így tovább.

A **D** operátor nem csak egyváltozós függvényekre alkalmazható. A parciális deriváltak a **D**-re vonatkozó indexelt függvényhívásokkal érhetők el:

```
> h := (x,y,z) -> 1/(x^2+y^2+z^2)^(1/2);
```

$$h := (x, y, z) \to \frac{1}{\sqrt{x^2+y^2+z^2}}$$

```
> # partial derivative w.r.t. x
> 'D[1](h)' = D[1](h);
```

$$D_1(h) = ((x, y, z) \to -\frac{x}{(x^2+y^2+z^2)^{3/2}})$$

`D[1](h)` a `h` függvény első argumentuma szerinti parciális deriváltját jelenti. Ez ekvivalens az `unapply( diff(h(x,y,z),x), x,y,z )` kifejezéssel, az esetleges eltérések az egyszerűsítésekből adódhatnak.

```
> # partial derivative w.r.t. x and y
> 'D[1,2](h)' = D[1,2](h);
```

$$D_{1,2}(h) = ((x, y, z) \to 3\,\frac{y\,x}{(x^2+y^2+z^2)^{5/2}})$$

`D[1,2](h)` ugyanazt jelenti, mint `D[1](D[2](h))`.

```
> # 2nd partial derivative w.r.t. x
> 'D[1,1](h)' = D[1,1](h);
```

$$D_{1,1}(h) = ((x, y, z) \to 3\,\frac{x^2}{(x^2+y^2+z^2)^{5/2}} - \frac{1}{(x^2+y^2+z^2)^{3/2}})$$

Ha a $\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) h$ Laplace-operátort alkalmazzuk, az egyszerűsítések elmaradása miatt nyilvánvalóvá válnak az operátor-módszer korlátjai:

```
> L[h] := ( D[1,1] + D[2,2] + D[3,3] )(h);
```

$$L_h := ((x, y, z) \to 3\,\frac{x^2}{(x^2+y^2+z^2)^{5/2}} - \frac{1}{(x^2+y^2+z^2)^{3/2}})$$
$$+ ((x, y, z) \to 3\,\frac{y^2}{(x^2+y^2+z^2)^{5/2}} - \frac{1}{(x^2+y^2+z^2)^{3/2}})$$
$$+ ((x, y, z) \to 3\,\frac{z^2}{(x^2+y^2+z^2)^{5/2}} - \frac{1}{(x^2+y^2+z^2)^{3/2}})$$

```
> normal( L[h](x,y,z) );
```

$$0$$

Minden út Rómába vezet. Bár az eddigiekben az implicit differenciálás végrehajtására a **diff** és a **D** eljárásokat használtuk, ugyanerre a célra a Maple könyvtár is tartalmaz egy **implicitdiff** nevű rutint. Ezzel igyekeznek megkönnyíteni a szoftver használóinak életét. Az [53] irodalom 1.5.4. fejezetében a MACSYMA rendszerrel megoldott példán mutatjuk be az eljárást. Legyen $g(x, y)$ olyan implicit formula, amelyben y az x függvénye. A továbbiakban y x szerinti első, másod és harmadrendű deriváltjára adunk meg képleteket a g parciális deriváltjainak segítségével.

```
> dydx := implicitdiff( g(x,y), y(x), x );
```

$$dydx := -\frac{D_1(g)(x, y)}{D_2(g)(x, y)}$$

Lehet, hogy az Olvasónak ismerősebb ez a jelölésmód:

```
> convert( dydx, diff );
```

$$-\frac{\frac{\partial}{\partial x}\mathrm{g}(x, y)}{\frac{\partial}{\partial y}\mathrm{g}(x, y)}$$

Folytassuk y magasabb rendű deriváltjainak meghatározásával:

```
> d2ydx2 := implicitdiff( g(x,y), y(x), x$2 );
```

$$d2ydx2 := -(D_{1,1}(g)(x, y)\, D_2(g)(x, y)^2 - 2\, D_{1,2}(g)(x, y)\, D_1(g)(x, y)\, D_2(g)(x, y) + D_{2,2}(g)(x, y)\, D_1(g)(x, y)^2) \Big/ D_2(g)(x, y)^3$$

```
> convert( ", diff );
```

$$-((\frac{\partial^2}{\partial x^2}\mathrm{g}(x, y))\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^2 - 2\,(\frac{\partial^2}{\partial y\,\partial x}\mathrm{g}(x, y))\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))\,(\frac{\partial}{\partial y}\mathrm{g}(x, y)) + (\frac{\partial^2}{\partial y^2}\mathrm{g}(x, y))\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))^2) \Big/ (\frac{\partial}{\partial y}\mathrm{g}(x, y))^3$$

y harmadik deriváltjának kiszámítása már jól mutatja a számítógépes algebra használatának előnyeit a hagyományos papírral és ceruzával szemben:

```
> d3ydx3 := convert( implicitdiff( g(x,y), y(x), x$3 ),
> diff );
```

$$\begin{aligned} d3ydx3 := -((&\frac{\partial^3}{\partial x^3}\mathrm{g}(x, y))\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^4 \\ &+ 3\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))^2\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^2\,(\frac{\partial^3}{\partial y^2\,\partial x}\mathrm{g}(x, y)) - 3\,\%2\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^3\,\%3 \\ &+ 6\,\%2^2\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^2\,(\frac{\partial}{\partial x}\mathrm{g}(x, y)) \\ &- 3\,(\frac{\partial^3}{\partial y\,\partial x^2}\mathrm{g}(x, y))\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^3 \\ &- (\frac{\partial}{\partial x}\mathrm{g}(x, y))^3\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))\,(\frac{\partial^3}{\partial y^3}\mathrm{g}(x, y)) \\ &+ 3\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))\,\%1\,\%3\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))^2 + 3\,\%1^2\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))^3 \\ &- 9\,\%2\,(\frac{\partial}{\partial y}\mathrm{g}(x, y))\,\%1\,(\frac{\partial}{\partial x}\mathrm{g}(x, y))^2) \Big/ (\frac{\partial}{\partial y}\mathrm{g}(x, y))^5 \end{aligned}$$

$$\%1 := \frac{\partial^2}{\partial y^2}\mathrm{g}(x, y)$$

$$\%2 := \frac{\partial^2}{\partial y\, \partial x}\, \mathrm{g}(x,\, y)$$

$$\%3 := \frac{\partial^2}{\partial x^2}\, \mathrm{g}(x,\, y)$$

Befejezésül kiszámoljuk a $g(x, y) = exp(x^2 + y^2)$ függvény deriváltjait:

```
> g := (x,y) -> exp( x^2 + y^2 );
```

$$g := (x,\, y) \to e^{(x^2+y^2)}$$

```
> dydx;
```

$$-\frac{x}{y}$$

```
> normal( d2ydx2 );
```

$$-\frac{x^2+y^2}{y^3}$$

```
> normal( d3ydx3 );
```

$$-3\,\frac{x\,(x^2+y^2)}{y^5}$$

Ezek a képletek összhangban vannak az $x^2 + y^2 = c$ egyenlettel definiált y implicit függvényre korábban kapott eredményekkel.

9.2. Automatikus differenciálás

A **D** operátor fölhasználható *automatikus differenciálásra*, vagyis Maple eljárások differenciálására is. Kezdjük egy olyan példával, amely case utasítással megadott szakaszonként definiált függvényt tartalmaz:

```
> F := x -> if x>0 then sin(x) else arctan(x) fi;
```

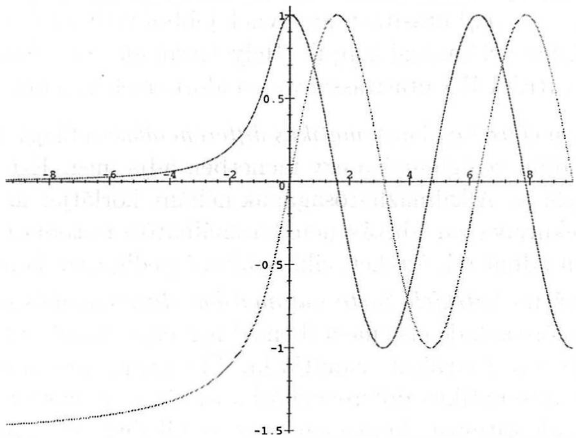
$F :=$ **proc**(x)
  option *operator*, *arrow*;
    if $0 < x$ **then** $\sin(x)$ **else** $\arctan(x)$ **fi**
  end

```
> Fp := D(F); # 1st derivative
```

$Fp :=$ **proc**(x)
  option *operator*, *arrow*;
    if $0 < x$ **then** $\cos(x)$ **else** $1/(1+x^2)$ **fi**
  end

Rajzoltassuk ki F-et első deriváltjával együtt.

```
> plot( { F, Fp }, -3*Pi..3*Pi );
```



9.1. ábra: Szakaszonként definiált függvénynek és deriváltjának grafikonja

A Maple több utasításból álló eljárásokat is tud differenciálni. Lássunk egy egyszerű példát:

```
> f := proc(x)
>   local s,t;
>   s := ln(x);
>   t := x^2;
>   s*t + 3*t
> end:
> fp := D(f);
```

$fp :=$ **proc**(x)
local $tx,\ sx,\ t,\ s;$
$sx := 1/x\,;\ s := \ln(x)\,;\ tx := 2 \times x\,;\ t := x^2\,;\ sx \times t + s \times tx + 3 \times tx$
end

Az **fp** eljárás valóban f'-t számolja ki? Ezt úgy bizonyíthatjuk be, hogy az eljárást szimbolikus paraméterekkel hívjuk meg, ami ténylegesen az eljárás által leírt függvény képletté alakítását jelenti.

```
> diff(f(x),x) - fp(x);
```

$$0$$

Ez a fajta ellenőrzés *nem* lehetséges olyan eljárásokra, amelyekben formális paramétert tartalmazó föltételes utasítás fordul elő (ilyen volt az előző F függvény is). Mellesleg ez az egyik oka az automatikus differenciálás bevezetésének: nem minden függvény írható le (kényelmesen) képlettel.

Hogyan konstruálható meg az **fp** eljárás? **f** és **fp** forráskódjának összehasonlítása kulcsot ad az automatikus differenciálás következő általános módszerének megértéséhez:

> Az eljárásban szereplő minden $v := f(v_1, \ldots, v_n)$ alakú értékadó utasítás elé (ahol v_i lokális változó vagy formális paraméter) szúrjuk be a $v_x := fp(v_1, \ldots, v_n)$ utasítást, amelynek jobboldalát az $f(v_1, \ldots, v_n)$ formális differenciálásával kapjuk. Helyettesítsük az utolsó utasítást (és minden RETURN utasítással visszaadott értéket) a deriváltjával.

A módszer neve *előre tartó automatikus differenciálás*; a függvényt és deriváltját az eljárás kódján végighaladva egy menetben adja meg. Ezt az algoritmust használja a Maple is. Alkalmazhatóságának néhány korlátja: az eljárás törzsében nem lehet rekurzív eljáráshívás, nem használható a `remember` opció, globális változóknak nem adhatunk értéket, ciklusváltozó pedig csak konstans lehet.

Létezik egy másik, *hátrafelé tartó automatikus differenciálásnak* hívott módszer, amely egy előre haladó első menetben a függvényt, majd egy hátrafelé tartó második menetben a deriváltat számítja ki. Ezt (még) nem implementálták a Maple-ben. Az automatikus differenciálásba történő bevezetést tartalmaz [87] és [88]. Sokféle algoritmust, implementációt és alkalmazást leíró jó kézikönyv [89].

Ezt a pontot azzal zárjuk le, hogy rámutatunk az automatikus differenciálás használatának másik okára. Ez a számítási idő és a memóriahasználat költségkihatása. Tegyük föl, hogy az

$$f_1 = x, f_n = x^{f_{n-1}} \text{ ha } n > 1$$

rekurziós összefüggéssel definiált iterált hatványokra és deriváltjaikra vagyunk kíváncsiak. Amint a 9.1. alfejezetben korábban láttuk, a deriváltak kifejezései már kis n értékekre meglehetősen terjengőssé válnak. A deriváltra automatikus differenciálással kapott eljárás viszont nagyon tömör. Az f_n következő iteratív definícióját fogjuk fölhasználni.

```
> f := proc(x,n)
>   local i,t;
>   t := 1;
>   for i to n do t := x^t od;
>   t
> end;
```

$$f := \mathbf{proc}(x, n)\, \mathbf{local}\, i, t;\ t := 1\, ;\ \mathbf{for}\, i\, \mathbf{to}\, n\, \mathbf{do}\, t := x^t\ \mathbf{od}\, ;\ t\, \mathbf{end}$$

```
> fp := D[1](f); # 1st derivative of f w.r.t. x
```

$fp := \mathbf{proc}(x, n)$
$\quad \mathbf{local}\ tx, i, t;$
$\quad\quad tx := 0\,;$
$\quad\quad t := 1\,;$
$\quad\quad \mathbf{for}\ i\ \mathbf{to}\ n\ \mathbf{do}\ tx := x^t \times (tx \times \ln(x) + t/x)\,;\ t := x^t\ \mathbf{od}\,;$
$\quad\quad tx$
$\quad \mathbf{end}$

Kiszámítjuk f_{22} harmadik deriváltját kétféle módon: automatikus differenciálással és szimbolikus differenciálással. Látható lesz a memóriahasználatban és a számítási időben mutatkozó nagy eltérés.

- automatikus differenciálás

```
> fppp := D[1$3](f): # 3rd derivative of f w.r.t. x
> # register memory usage
> setbytes := kernelopts( bytesused ):
> # start timing
> settime := time():
> fppp( 1.1, 22 );
```

$$18.23670379$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := .290\,seconds$$

```
> memory_used := evalf(
> ( kernelopts( bytesused ) - setbytes ) /
> 1024 * kbytes, 5);
```

$$memory_used := 426.17\,kbytes$$

- formális differenciálás[1]

```
> f22 := unapply( f(x,22), x );
```

$$f22 := x \to x^{\left(x^{\left(x^{\left(x^{\left(x^{(\dots)}\right)}\right)}\right)}\right)}$$

```
> setbytes := kernelopts( bytesused ):
> settime := time():
> (D@@3)(f22)(1.1);
```

$$18.23670379$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 63.959\,seconds$$

```
> memory_used := evalf(
> ( kernelopts( bytesused ) - setbytes ) /
> 1024 * kbytes, 5);
```

$$memory_used := 40997.\,kbytes$$

[1] Az unapply eredményét az olvashatóság érdekében egyszerűsítve közöljük. (*A Fordító megjegyzése.*)

9.3. Gyakorlatok

1. Legyenek f, g és h a következő többváltozós valós függvények:

$$
\begin{aligned}
f(x,y) &:= \frac{\ln(1+x^4+y^4)}{\sqrt{x^2+y^2}} \\
g(x,y,z) &:= \frac{1}{\sqrt{(x-a)^2+(y-b)^2+(z-c)^2}} \\
h(x,y,z) &:= \frac{z}{x^2+y^2+z^2}.
\end{aligned}
$$

(a) Határozzuk meg f összes másodrendű parciális deriváltját.

(b) Ellenőrizzük, hogy g a Laplace-egyenlet megoldása, vagyis

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right) g = 0$$

(c) Ellenőrizzük, hogy h a

$$\frac{\partial^2 h}{\partial x \partial y} + \left(\frac{4x}{x^2+y^2+z^2}\right)\frac{\partial h}{\partial y} = 0$$

differenciálegyenlet megoldása.

2. Hasonlítsuk össze a következő Maple parancsok eredményét:

```
> diff( f(x), x );
> convert( ", 'D' );
> unapply( ", x );
```

3. Határozzuk meg az $f(x) := \max(x^3, x)$ függvény deriváltját.

4. Jelölje $y(x)$ a $\sqrt{x}+\sqrt{y}=1$ egyenlettel meghatározott implicit függvényt. Számítsuk ki az y' első és az y'' második deriváltat.

5. Valamely h háromváltozós függvényre jelölje $z(x,y)$ a $h(x,y,z)=0$ egyenlettel meghatározott implicit függvényt. Állítsuk elő a $\dfrac{\partial z}{\partial x}$ és a $\dfrac{\partial^2 z}{\partial x \partial y}$ deriváltak képletét. Hogyan néznek ki az eredmények, ha $h = \sqrt{x}+\sqrt{y}+\sqrt{z}-1$?

6. Tekintsük az alábbi rekurzióval definiált f_n függvényt:

$$f_0 = 0, \quad f_1 = x, \quad f_n = f_{n-1} + \sin(f_{n-2}) \text{ ha } n > 1.$$

Határozzunk meg (automatikus differenciálással) eljárást az f_n első deriváltjának kiszámítására.

10.

Integrálás és összegzés

A (határozatlan és a határozott) integrálás a számítógépes algebra egyik fontos területe. A Maple az elemi függvények integrálására a legtöbb matematikai kézikönyvben tárgyalt heurisztikus integrálási módszerek helyett olyan nem klasszikus algoritmusokat használ, mint a Risch-féle algoritmus. Röviden ismertetjük a Maple integrálok meghatározására alkalmazott stratégiáját, és sok példát közlünk, hogy képet kapjunk a Maple lehetőségeiről, valamint arról, hogyan segíthetünk a rendszernek. Mutatunk példákat integráltranszformációkra (a Laplace, a Fourier, a Mellin, a Hilbert és a Mellin-félére) is.

Az integrálás diszkrét megfelelőjének tekinthető összegzés az analízis másik olyan fontos területe, ahol a Maple fejlett módszereket alkalmaz. Ezt néhány példával fogjuk szemlélteni.

10.1. Határozatlan integrálok kiszámítása

A Maple-nek több hatékony, beépített függvényintegrálási algoritmusa van. Először a középiskolában és az egyetemen tanított hagyományos technikákat próbálja alkalmazni: táblázatokban keres, mintaillesztéssel, parciális integrálással, új változó bevezetésével, a láncszabály alkalmazásával stb. próbálkozik. Ha ezek a heurisztikus módszerek nem vezetnek eredményre, akkor a rendszer determinisztikus módszereket alkalmaz, főleg az úgynevezett Risch-féle algoritmust [163]. Tekintsünk először egy példát az **int** eljárással végzett határozatlan integrálásra:

```
> Int( x/(x^3+1), x ) = int( x/(x^3+1), x );
```

$$\int \frac{x}{x^3+1}\,dx = -\frac{1}{3}\ln(1+x) + \frac{1}{6}\ln(x^2-x+1) + \frac{1}{3}\sqrt{3}\arctan(\frac{1}{3}(2x-1)\sqrt{3})$$

Figyeljük meg, hogy a Maple elhagyta az integrációs konstanst. A kapott eredmények így könnyebben kezelhetők. Egy kis óvatosság azért nem árt; ellenőrizzük a rendszer válaszát.

```
> diff( rhs("), x );
```

$$-\frac{1}{3}\frac{1}{1+x}+\frac{1}{6}\frac{2x-1}{x^2-x+1}+\frac{2}{3}\frac{1}{1+\frac{1}{3}(2x-1)^2}$$

```
> normal( ", 'expanded' );
```

$$\frac{x}{x^3+1}$$

A fejezet további részében találkozunk majd olyan példákkal is, ahol az integrálás eredménye nem ellenőrizhető ilyen könnyedén.

A fönti példa baloldalán a megjelenítés kedvéért az **Int** tétlen eljárást alkalmaztuk. Ezentúl az **Int**-et fogjuk használni az integrandus és az integrálási határok megmutatására, és ténylegesen a **value** eljárással számíttatjuk ki az integrált.

A racionális függvények integrálására az iskolában tanult módszerek közül a parciális törtekre bontás a legfontosabb. Jól alkalmazható a fönti integrálra is. A következőhöz hasonló esetekre való alkalmazása azonban már jóval fárasztóbb, és több hibalehetőséggel jár:

```
> Int( x/(x^5+1), x ):  " = value(");
```

$$\int \frac{x}{x^5+1}\,dx = -\frac{1}{5}\ln(1+x)+\frac{1}{20}\ln(2x^2-x-\sqrt{5}x+2)$$
$$-\frac{1}{20}\ln(2x^2-x-\sqrt{5}x+2)\sqrt{5}+\frac{2}{5}\frac{\arctan\left(\frac{4x-1-\sqrt{5}}{\sqrt{10-2\sqrt{5}}}\right)\sqrt{5}}{\sqrt{10-2\sqrt{5}}}$$
$$+\frac{1}{20}\ln(2x^2-x+\sqrt{5}x+2)+\frac{1}{20}\ln(2x^2-x+\sqrt{5}x+2)\sqrt{5}$$
$$-\frac{2}{5}\frac{\arctan\left(\frac{4x-1+\sqrt{5}}{\sqrt{10+2\sqrt{5}}}\right)\sqrt{5}}{\sqrt{10+2\sqrt{5}}}$$

```
> normal( diff(rhs("),x), 'expanded' ); # check the answer
```

$$\frac{x}{x^5+1}$$

És ha csak egy kicsit megváltoztatjuk az integrandust, már nem is alkalmazható a parciális törtekre bontás trükkje:

```
> infolevel[int] := 2:  # make Maple more communicative
> Int( x/(x^5+2*x+1), x ):  " = value(");
```

```
int/indef:   first-stage indefinite integration
int/ratpoly:   rational function integration
int/rischnorm:   enter Risch-Norman integrator
int/risch:   enter Risch integration
int/risch:   the field extensions are
```

$$[_X]$$

```
unknown:   integrand is
```

$$\frac{_X}{_X^5 + 2\,_X + 1}$$

```
int/risch/ratpoly:   integrating
```

$$\frac{_X}{_X^5 + 2\,_X + 1}$$

```
int/risch/ratpoly:   Horowitz' method yields
```

$$\int \frac{_X}{_X^5 + 2\,_X + 1}\, d_X$$

```
int/risch/ratpoly:
starting computing subresultants at time   1.830
int/risch/ratpoly:
end of subresultants computation at time   1.840
int/risch/ratpoly:
Rothstein's method - factored resultant is
```

$$[[z^5 - \frac{500}{11317} z^3 + \frac{4}{11317} z + \frac{1}{11317}, 1]]$$

```
int/risch/ratpoly:   result is
```

$$\sum_{_R=\%1} _R \ln(_X + \frac{65376045600}{64828679}\,_R^4 - \frac{4270752875}{64828679}\,_R^3 - \frac{625000000}{64828679}\,_R^2 + \frac{447235682}{64828679}\,_R - \frac{21514240}{64828679})$$
$$\%1 := \mathrm{RootOf}(11317\,_Z^5 - 500\,_Z^3 + 4\,_Z + 1)$$

```
int/risch:   exit Risch integration
```

$$\int \frac{x}{x^5 + 2\,x + 1}\, dx = \sum_{_R=\%1} _R \ln(x + \frac{65376045600}{64828679}\,_R^4 - \frac{4270752875}{64828679}\,_R^3 - \frac{625000000}{64828679}\,_R^2 + \frac{447235682}{64828679}\,_R - \frac{21514240}{64828679})$$
$$\%1 := \mathrm{RootOf}(11317\,_Z^5 - 500\,_Z^3 + 4\,_Z + 1)$$

```
>  normal( diff(rhs("),x), 'expanded' ); # check the answer
```

$$\frac{x}{x^5 + 2\,x + 1}$$

```
> infolevel[int] := 0:  # back to default information level
```

Az előző példában az infolevel[int] értékét 2-re állítottuk be, így látható, hogy a Maple milyen módszert használ. A Maple racionális függvények integrálására a következő általános megközelítést alkalmazza.

- A határozatlan integrálás „első lépcsőjében" a Maple a táblázatokban való kereséshez hasonló, egyszerű módszerekkel kísérletezik, és az integrandus formája alapján eldönti, hogy a továbbiakban melyik módszert kellene még kipróbálni.
- Ha az egyszerű módszerek nem vezetnek eredményre, elkezdődik a „második lépcső". Ebben a Maple heurisztikus módszerekkel próbálkozik: derivált-osztással (az integrandus $\frac{p}{q}$ alakú-e, ahol q' osztja p-t), helyettesítésekkel és (hétnél alacsonyabb fokszámú nevezőkre) parciális törtekre bontással. Továbbá megvizsgálja az olyen speciális formájú integrandusokat, mint az $f(ux+v)\,\frac{p(x)}{q(x)}$ vagy $Bessel(x)\,p(x)$, amelyekben f az exp, ln, sin, cos, sinh és a cosh függvények valamelyike, p és q pedig polinomok.
- Ha a második lépcső heurisztikus módszerei kudarcot vallottak, a Maple a Risch-féle algoritmusba kezd. Racionális függvényekre ez a következőképpen történik.

  (i) A [105] szerinti Horowitz-redukciót alkalmaz az integrál $\frac{c}{d}+\int\frac{a}{b}$ alakba történő átírásához, ahol a, b, c és d polinomok, b négyzetmentes és a fokszáma kisebb, mint b fokszáma. Ebben az alakban a $\frac{c}{d}$-t racionális résznek hívják, mivel a maradék integrál kifejezhető csak logaritmusok segítségével. Ezért az $\int\frac{a}{b}$ kifejezést logaritmusos résznek szokás nevezni.

  (ii) A Rothstein–Trager-féle módszerrel (lásd [164, 182]) a Maple a logaritmusos részt $\sum_i c_i \log v_i$ alakúra hozza, ahol a c_i-k nemnulla konstansok, a v_i-k pedig pozitív fokszámú relatív prím négyzetmentes főpolinomok. A Lazard–Rioboo–Trager-féle javítást implementálták a rendszerben, amely a logaritmusos rész számításánál elkerüli az algebrai bővítéseket és faktorizációkat.

Természetesen megkérdőjelezhető az utolsó válasz használhatósága, mivel egy ötödfokú polinom analitikusan kiszámíthatatlan zérushelyeit tartalmazza. De ugyanez az algoritmus eredményre vezet a [181]-ből vett következő példához hasonló esetekben.

```
> Int( (7*x^13+10*x^8+4*x^7-7*x^6-4*x^3-4*x^2+3*x+3) /
> (x^14-2*x^8-2*x^7-2*x^4-4*x^3-x^2+2*x+1), x ):
> " = value(");
```

$$\int\frac{7\,x^{13}+10\,x^8+4\,x^7-7\,x^6-4\,x^3-4\,x^2+3\,x+3}{x^{14}-2\,x^8-2\,x^7-2\,x^4-4\,x^3-x^2+2\,x+1}\,dx =$$
$$\frac{1}{2}\ln(x^7-\sqrt{2}\,x^2+(-\sqrt{2}-1)\,x-1)\,\sqrt{2}$$

$$+\frac{1}{2}\ln(x^7-\sqrt{2}\,x^2+(-\sqrt{2}-1)\,x-1)$$
$$+\frac{1}{2}\ln(x^7+\sqrt{2}\,x^2+(\sqrt{2}-1)\,x-1)$$
$$-\frac{1}{2}\ln(x^7+\sqrt{2}\,x^2+(\sqrt{2}-1)\,x-1)\sqrt{2}$$

```
> normal( diff(rhs("),x), 'expanded' ); # check the answer
```

$$\frac{7\,x^{13}+10\,x^8+4\,x^7-7\,x^6-4\,x^3-4\,x^2+3\,x+3}{x^{14}-2\,x^8-2\,x^7-2\,x^4-4\,x^3-x^2+2\,x+1}$$

A Risch-féle algoritmus [163] ereje akkor válik nyilvánvalóvá, ha egy tágabb függvényosztályra, nevezetesen az *elemi függvények* osztályára alkalmazzuk. Az x változó elemi függvényeinek hevenyészett definíciója:

- Induljunk ki konstansok valamely halmazából (ez lehet például $\mathbb{Q}$, $\mathbb{R}$ vagy $\mathbb{C}$).
- Konstruáljuk meg az összes konstans együtthatós $p(x)$ polinomot.
- Konstruáljuk meg a $\dfrac{p(x)}{q(x)}$ alakú racionális törtfüggvényeket.
- Vegyük mindezek exponenciális kifejezéseit (ebbe bele fognak tartozni a trigonometrikus és a hiperbolikus függvények, valamint ezek inverzei is, ha az i komplex szám előfordul a konstansok között).
- Bővítsünk az eddig kapott függvények logaritmusaival (ezzel az *elemi transzcendens függvényeket* kapjuk).
- Bővítsünk az algebrai függvényekkel, vagyis az olyan polinomiális egyenletek megoldásaival, amelyeknek együtthatói a korábban bevezetett típusú függvények (ilyen például a $\sqrt{x^2+1}$, amely az $y^2-x^2-1=0$ egyenlet megoldása).

A konstrukció minden lépésében (a konstansok választását kivéve) megengedett az iteráció is, így képezhetjük logaritmusok polinomjainak racionális függvényét stb. .

Risch [163]-ban egy olyan algoritmust írt le, amely adott elemi transzcendens függvényről eldönti, hogy az integrálja kifejezhető-e elemi függvényként, és ha igen, kiszámítja az integrált. Ez az algoritmus a következő tételen alapul:

Liouville-elv *Ha az $f(x)$ elemi függvény integrálja is elemi függvény, akkor*

$$\int f(x)\,dx = v(x)+\sum_{i=1}^{n} c_i\log(u_i(x))$$

alakú, ahol a c_i-k konstansok, és a v meg az u_i függvények nem tartalmaznak f-ben és a c_i konstansokban elő nem forduló mennyiségeket.

Liouville tétele tehát bizonyos értelemben az elemi integrálok keresési terét korlátozza. Tovább általánosításként vehetnénk a *Liouville-féle függvények* osztályát, amelyet az elemi függvényekből úgy konstruálunk meg, hogy az osztályhoz tartozó tetszőleges $f(x)$ függvénnyel fölírható $\int f(x)\,dx$ integrálokkal bővítünk (ez a konstrukció is iterálható). Az így kapott osztály tartalmazza például az erf hibafüggvényt és az Ei exponenciális integrálfüggvényt, amelyeket a következő képletek definiálnak:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\,dt \quad \text{és} \quad \operatorname{Ei}(x) = \int_x^\infty \frac{\exp(-t)}{t}\,dt.$$

A Risch-féle algoritmus részletei után érdeklődő Olvasó további információkat a [20,21,52,53,74,76,143,163] munkákban találhat. Itt csak röviden vázoljuk, hogy a Maple általában hogyan végzi az elemi függvények integrálását:

- Ha a heurisztikus módszerek sikertelenek, akkor a Maple a „Risch–Norman előtétet" (lásd [74]) alkalmazza, hogy trigonometrikus és hiperbolikus függvényeket tartalmazó integrandusokhoz ne kelljen bevezetni komplex exponenciális és logaritmus függvényeket.
- Ha továbbra sincs eredmény, akkor elkezdődik a Risch-algoritmus.

Az utolsó lépés legnehezebb része az algebrai függvények integrálása. A Trager-féle módszert [118] az algebrai függvények **RootOf**-os jelölésével implementálták. A jelenlegi megvalósítás azonban néhány korlátozást tartalmaz:

- Ha az együtthatók testje nem algebrai számtest, az integrál transzcendens részét nem számítja ki a rendszer. A gyakorlatban ez azt jelenti, hogy ha az integrál paramétert tartalmaz, akkor előfordulhat, hogy a Maple nem találja meg a választ, jóllehet az létezik.
- Az algoritmus bonyolultsága miatt lassú.
- Az integrandusnak az integrálási változó egyetlen **RootOf** kifejezését kell tartalmaznia. Az **evala@Primfield** eljárás használható föl ennek a föltételnek az elérésére.

Vizsgáljunk meg néhány példát, hogy pontosabb képet kapjunk a Maple integrálási képességeiről:

```
> Int( ln(x-1)^3, x ):  " = value(");
```

$$\int \ln(x-1)^3\,dx = \ln(x-1)^3\,(x-1) - 3\,(x-1)\ln(x-1)^2 + 6\,(x-1)\ln(x-1) + 6 - 6\,x$$

```
> diff( rhs("), x ); # check the answer
```

$$\ln(x-1)^3$$

```
> int( ln(x-1)^2/x, x );
```

$$\int \frac{\ln(x-1)^2}{x}\, dx$$

Az első integrált parciális integrálással és táblabeli kereséssel számította ki a Maple integráló eljárásának előtétje. A Risch-algoritmussal eldöntötte, hogy a második integrál nem fejezhető ki zárt alakban elemi függvényekkel. Ilyenkor a Maple vagy csak egyszerűen visszhangozza az input parancsot, vagy új függvényeket vezet be (mint például a hibafüggvény, az exponenciális integrál, a szinusz integrál stb.).

```
> Int( exp(-x^2), x ):  " = value(");
```

$$\int e^{(-x^2)}\, dx = \frac{1}{2}\sqrt{\pi}\,\mathrm{erf}(x)$$

```
> Int( exp(-x)/x, x ):  " = value(");
```

$$\int \frac{e^{(-x)}}{x}\, dx = -\mathrm{Ei}(1, x)$$

A heurisztikus módszereken kívűl a csak logaritmusokat tartalmazó integrandusok esetén a Maple a következő utat választja:

```
> Int( ln(1-b*x/(a+c*x^2))/x, x ):  " = value(");
```

$$\int \frac{\ln(1-\frac{b\,x}{a+c\,x^2})}{x}\, dx = \ln(x)\ln(1-\frac{b\,x}{a+c\,x^2})$$
$$-\ln(\frac{1}{2}\frac{b+\sqrt{b^2-4\,c\,a}}{c})\ln(-\frac{-2\,c\,x+b+\sqrt{b^2-4\,c\,a}}{b+\sqrt{b^2-4\,c\,a}})$$
$$-\ln(-\frac{1}{2}\frac{-b+\sqrt{b^2-4\,c\,a}}{c})\ln(-\frac{2\,c\,x-b+\sqrt{b^2-4\,c\,a}}{-b+\sqrt{b^2-4\,c\,a}})$$
$$+\mathrm{dilog}(2\frac{c\,x}{b+\sqrt{b^2-4\,c\,a}})+\mathrm{dilog}(-2\frac{c\,x}{-b+\sqrt{b^2-4\,c\,a}})$$
$$+\ln(\frac{\sqrt{-c\,a}}{c})\ln(\frac{c\,x-\sqrt{-c\,a}}{\sqrt{-c\,a}})+\ln(-\frac{\sqrt{-c\,a}}{c})\ln(-\frac{c\,x+\sqrt{-c\,a}}{\sqrt{-c\,a}})$$
$$-\mathrm{dilog}(\frac{c\,x}{\sqrt{-c\,a}})-\mathrm{dilog}(-\frac{c\,x}{\sqrt{-c\,a}})$$

```
> Int( sin(x) / ( x^3 + x + 1 ), x ):  " = value(");
```

$$\int \frac{\sin(x)}{x^3+x+1}\, dx =$$
$$\sum_{_R1=\%1} \frac{\mathrm{Si}(x-_R1)\cos(_R1)+\mathrm{Ci}(x-_R1)\sin(_R1)}{3\,_R1^2+1}$$
$$\%1 := \mathrm{RootOf}(_Z^3+_Z+1)$$

Az előző esetben a Maple észrevette, hogy az integrandus egy szinusz függvényt tartalmaz, és a Hermite-féle redukciót alkalmazta a pusztán trigonometrikus függvényekből álló kifejezésekre.

```
> Int( 1 / ( x * (x^2+1)^(1/3) ), x ):  " = value(");
```

$$\int \frac{1}{x\,(x^2+1)^{1/3}}\,dx = \int \frac{1}{x\,(x^2+1)^{1/3}}\,dx$$

Ebben az esetben a Maple ragaszkodik a radikálok **RootOf**-os jelöléséhez. Jelölje α az x^2+1 köbgyökét. Az eredményből kitűnik, hogy a z^2+z+1 polinom gyökére célszerű a β rövidítést bevezetni:

```
> alias( alpha = RootOf( z^3 - x^2 - 1, z ),
>   beta = RootOf( z^2 + z + 1, z ) ):
> convert( "", RootOf ):
```

Ezekkel a jelölésekkel már föl tudja írni az integrált a Maple:

```
> settime := time():  # start timing
> ""; # evaluation
```

$$\int \frac{1}{x\,\alpha}\,dx = \frac{1}{2}\ln(-(5 - 19\,\beta - 4\,\beta^2 + 2\,x^2 - 9\,\beta\,x^2 + 4\,\beta^2\,x^2 - 9\,\alpha\,\beta$$
$$- 24\,\alpha + 15\,\alpha^2 + 24\,\alpha^2\,\beta)\,/x^2) + \frac{1}{2}\beta\ln(($$
$$17\,\beta - 4 - 4\,\beta^2 + 4\,\beta^2\,x^2 + 11\,\beta\,x^2 - 3\,x^2 + 21\,\alpha - 21\,\alpha^2 - 21\,\alpha^2\,\beta)$$
$$/x^2)$$

```
> # check the answer
> evala( Normal( diff(rhs("),x) - 1/(x*alpha) ) );
```

$$0$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 87.610\,seconds$$

A [41] szerinti

$$\int \frac{2x^6 + 4x^5 + 7x^4 - 3x^3 - x^2 - 8x - 8}{(2x^2-1)^2\sqrt{x^4+4x^3+2x^2+1}}\,dx$$

Csebisev-integrált [52]-ben a REDUCE segítségével számolták ki. A Maple esetében a számítás folyamata és az eredmény a következő:

```
> alias( beta =
>   RootOf( z^2 - x^4 - 4*x^3 - 2*x^2 - 1, z ) ):
> settime := time():
> Int( ( 2*x^6 + 4*x^5 + 7*x^4 - 3*x^3 - x^2 - 8*x - 8) /
>   ( (2*x^2-1)^2 * beta ), x):  " = value(");
```

$$\int \frac{2\,x^6 + 4\,x^5 + 7\,x^4 - 3\,x^3 - x^2 - 8\,x - 8}{(2\,x^2-1)^2\,\beta}\,dx = \frac{1}{2}\,\frac{(2\,x+1)\,\beta}{2\,x^2-1} + \frac{1}{2}\ln(($$
$$1025\,x^{10} + 6138\,x^9 + 12307\,x^8 + 10188\,x^7 + 4503\,x^6 + 3134\,x^5$$
$$+ 1589\,x^4 + 140\,x^3 + 176\,x^2 + 2 - 4104\,\beta\,x^7 - 2182\,\beta\,x^5$$

$$- 5084\,\beta\,x^6 - 805\,\beta\,x^4 - 624\,\beta\,x^3 - 28\,\beta\,x - 10\,\beta\,x^2 - 1023\,\beta\,x^8)$$
$$/(2\,x^2 - 1)^5)$$

```
> # check the answer
> evala( Normal( diff(rhs("),x) - op(1,lhs(")) ) );
```

$$0$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 19.261\,seconds$$

A transzcendens rész kiszámítása igényli a legtöbb időt.

Algebrai függvényeket láthatunk a következő két integrálban:

```
> Int( x/(x^3 + x + a), x ):  " = value(");
```

$$\int \frac{x}{x^3 + x + a}\,dx = \sum_{_R=\%1} _R \ln(x + 9\,\frac{a\,(4 + 27\,a^2)\,_R^2}{2 + 27\,a^2} + \frac{(4 + 27\,a^2)\,_R}{2 + 27\,a^2} + 6\,\frac{a}{2 + 27\,a^2})$$
$$\%1 := \mathrm{RootOf}((4 + 27\,a^2)\,_Z^3 + _Z + a)$$

```
> Int( 1/(x^16 + a), x ):  " = value(");
```

$$\int \frac{1}{x^{16} + a}\,dx = \sum_{_R=\%1} _R \ln(x + 16\,a\,_R)$$
$$\%1 := \mathrm{RootOf}(18446744073709551616\,a^{15}\,_Z^{16} + 1)$$

A következő példa, amely nem más, mint az egyik előző integrál parametrizált változata, megmutatja az algebrai függvények integrálásának a jelenlegi implementációban meglévő korlátját, miszerint az együtthatók teste csak algebrai számtest lehet. Az outputban nagybetűsre változtattuk az erre vonatkozó üzeneteket:

```
> alias( gamma = RootOf( z^3 - x^2 - a, z ) ):
> infolevel[int] := 1:  # set higher information level
> int( 1/(x*gamma), x );

int/indef:   first-stage indefinite integration
int/indef2:   second-stage indefinite integration
int/rischnorm:   enter Risch-Norman integrator
int/risch:   enter Risch integration
int/algrisch/int:
Risch/Trager's algorithm for algebraic function
int/algrisch/int:
computation of the algebraic part: start time  114.250
int/algrisch/int:
computation of the algebraic part: end time  114.280
int/algrisch/int:
computation of the transcendental part: start time  114.299
int/algrisch/transcpar:   PARAMETRIC CASE NOT HANDLED YET
int/algrisch/int:
computation of the transcendental part: end time  114.310
```

```
int/algrisch/int:
could not find an elementary antiderivative
```

$$\int \frac{1}{x\,\gamma}\,dx$$

```
> infolevel[int] := 0:  # back to default value
```

10.2. Határozott integrálás

A Maple rendszerben a határozott integrálás is az **int**-tel, illetve a neki megfelelő **integrate** álnévvel végeztethető el. A megjelenítésre ismét a **value**-val kombinált **Int** tétlen formát használjuk:

```
> Int( x/(x^3+1), x = 1..a ):  " = value(");
```

$$\int_1^a \frac{x}{x^3+1}\,dx = -\frac{1}{3}\ln(1+a) + \frac{1}{6}\ln(a^2-a+1) + \frac{1}{3}\sqrt{3}\arctan(\frac{1}{3}\sqrt{3}\,(2\,a-1)) + \frac{1}{3}\ln(2) - \frac{1}{18}\sqrt{3}\,\pi$$

```
> Int( ln(t)/(1-t), t = 0..x ):  " = value(");
```

$$\int_0^x \frac{\ln(t)}{1-t}\,dt = \mathrm{dilog}(x) - \frac{1}{6}\,\pi^2$$

```
> Int( 1/((1+x^2)*(1+2*x^2)), x = 0..1 ):
> " = value(");
```

$$\int_0^1 \frac{1}{(1+x^2)\,(1+2\,x^2)}\,dx = -\frac{1}{4}\,\pi + \sqrt{2}\arctan(\sqrt{2})$$

A határozott integrálás nem mindig a megfelelő határozatlan integrál kiszámításával és a határok behelyettesítésével történik. Ezt mutatja az $\int_{-1}^{1} \frac{1}{x^2}\,dx$-re vonatkozó következő példa:

```
> Int( 1/x^2, x ):  " = value(");
```

$$\int \frac{1}{x^2}\,dx = -\frac{1}{x}$$

```
> subs( x=1, rhs(") ) - subs( x=-1, rhs(") );
```

$$-2$$

Itt az a probléma, hogy az analízis alaptételének alkalmazásához szükséges analitikus föltételek nem teljesülnek: az integrandusnak a $(-1,1)$ intervallum $x=0$ belső pontjában nem megszüntethető szingularitása van. A Maple ellenőrzi az adott intervallumon az integrandus folytonosságát, és hibás válasz lehetősége esetén egyszerűen visszaadja a parancsot. Mellesleg a Maple rákényszeríthető arra, hogy az integrandust folytonosnak tekintse, ha megadjuk a `continuous` kulcsszót. A választott példában maga a Maple is be tudja bizonyítani, hogy a határozott integrál divergens:

```
> Int( 1/x^2, x = -1..1 ):  " = value(");
```

$$\int_{-1}^{1} \frac{1}{x^2}\, dx = \infty$$

Nehezebb feladat az

$$\int_0^{2\pi} \frac{1}{1 + 3\, sin^2 t}\, dt$$

integrál meghatározása:

```
> Int( 1/(1+3*sin(t)^2), t = 0..2*Pi ):  " = value(");
```

$$\int_0^{2\,\pi} \frac{1}{1 + 3\sin(t)^2}\, dt = \pi$$

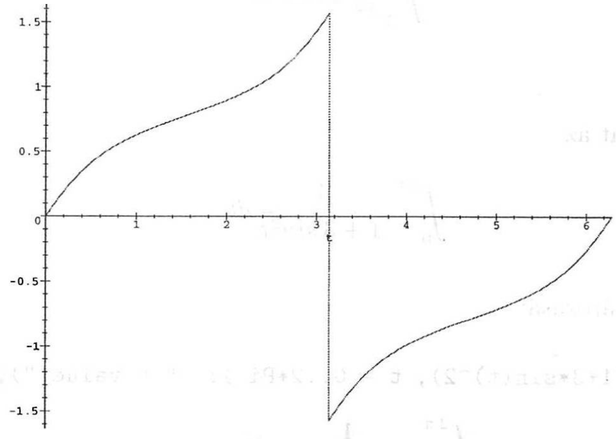
Az eredmény korrekt. De ez sem úgy jött ki, hogy kiszámoltuk a határozatlan integrált, és behelyettesítettük a határokat. A Maple által kiszámolt határozatlan integrál még csak nem is folytonos az integrálási intervallumon (lásd a 10.1. ábrát), bár létezik folytonos antiderivált is, a

$$\frac{t - \arctan\left(\dfrac{\sin(2t)}{\cos(2t) - 3}\right)}{2}.$$

```
> Int( 1/(1+3*sin(t)^2), t ):  " = value(");
```

$$\int \frac{1}{1 + 3\sin(t)^2}\, dt = 2\,\frac{\arctan\left(2\,\dfrac{\tan(\frac{1}{2}\,t)}{4 + 2\sqrt{3}}\right)}{4 + 2\sqrt{3}} + \frac{\arctan\left(2\,\dfrac{\tan(\frac{1}{2}\,t)}{4 + 2\sqrt{3}}\right)\sqrt{3}}{4 + 2\sqrt{3}}$$
$$- \frac{\arctan\left(2\,\dfrac{\tan(\frac{1}{2}\,t)}{4 - 2\sqrt{3}}\right)\sqrt{3}}{4 - 2\sqrt{3}} + 2\,\frac{\arctan\left(2\,\dfrac{\tan(\frac{1}{2}\,t)}{4 - 2\sqrt{3}}\right)}{4 - 2\sqrt{3}}$$

```
> plot( rhs("), t = 0..2*Pi );
```



10.1. ábra: Az $\frac{1}{1+3\,sin^2 t}$ nem folytonos antideriváltjának grafikonja

A határozott integrálok kiszámítására a Maple sokszor alkalmazza a táblabeli keresést, a mintaillesztést (lásd [73, 167]) és a speciális függvények paraméterek szerinti differenciálását.

```
> Int( exp(-sqrt(t)) / ( t^(1/4) * (1-exp(-sqrt(t))) ),
>    t = 0..infinity ): " = value(");
```

$$\int_0^\infty \frac{e^{(-\sqrt{t})}}{t^{1/4}\,(1-e^{(-\sqrt{t})})}\,dt = \sqrt{\pi}\,\zeta(\frac{3}{2})$$

```
> evalf( rhs(") );
```

$$4.630314748$$

```
> Int( t^4 * ln(t)^2 / (1+3*t^2)^3, t = 0..infinity ):
> " = value(");
```

$$\int_0^\infty \frac{t^4 \ln(t)^2}{(1+3\,t^2)^3}\,dt =$$
$$\frac{1}{216}\sqrt{3}\,\pi + \frac{1}{576}\,\pi^3\sqrt{3} - \frac{1}{108}\,\pi\sqrt{3}\ln(3) + \frac{1}{576}\,\pi\sqrt{3}\ln(3)^2$$

Ezek az integrálok az alábbi általános alakú integrál speciális esetei:

$$\int_0^\infty \frac{\exp(-u_1 t^{s_1} - u_2 t^{s_2})t^w \ln(b\,t^{d_l})^m \left\{ {\cos \atop \sin} \right\} (c\,t^r)}{(a_0 + a_1 f^d)^p}\,dt,$$

ahol

$f = t$ vagy $\exp(t^s)$,

p és m nemnegatív egészek,

$\text{signum}(a_0/a_1) > 0$,

s_1 és s_2 nemmulla valós számok,

d és d_l tetszőleges valós számok,

b, u_1 és u_2 pozitív valós számok,

w olyan komplex szám, amelyre $\Re\left(\frac{w+1}{s}\right) > 0$ teljesül, valamint

vagy $r = 0$, vagy $r = s$, vagy $r = 2s$, vagy $s = 2r$.

Mintaillesztéssel meghatározható további típusintegrálok:

$$\int_0^\infty \exp(-u\,t^s)t^w \ln(b\,t^{d_l})^m \begin{Bmatrix} \text{erf} \\ \text{erfc} \end{Bmatrix} (f\,t^v + g)\,dt,$$

$$\int_0^\infty \exp(-u\,t^s)t^w \text{BesselJ}(b,\, c\,t^{s_p})^{k_J}\, \text{BesselY}(\pm b,\, c\,t^{s_p})^{k_Y}\,dt,$$

és

$$\int_{t_0}^{t_1} \begin{Bmatrix} \sin \\ \cos \end{Bmatrix} \left(z \begin{Bmatrix} \sin \\ \cos \end{Bmatrix} \left(a\,t\right)\right) \begin{Bmatrix} \sin \\ \cos \end{Bmatrix} \left(b\,t\right) dt.$$

Két további példa:

```
> Int( t * exp(-t^2) * erf(2*t+1), t = 0..infinity ):
> " = value(");
```

$$\int_0^\infty t\,e^{(-t^2)}\,\text{erf}(2\,t+1)\,dt =$$
$$\frac{1}{2}\,\text{erf}(1) + \frac{1}{5}\,e^{(-1/5)}\,\sqrt{5} - \frac{1}{5}\,e^{(-1/5)}\,\sqrt{5}\,\text{erf}(\frac{2}{5}\,\sqrt{5})$$

```
> Int( x * exp(-x^2) * BesselJ(0,x) * BesselY(0,x),
> x = 0..infinity ):  " = value(");
```

$$\int_0^\infty x\,e^{(-x^2)}\,\text{BesselJ}(0,\,x)\,\text{BesselY}(0,\,x)\,dx = -\frac{1}{2}\,\frac{e^{(-1/2)}\,\text{BesselK}(0,\,\frac{1}{2})}{\pi}$$

```
> Int( sin(z*sin(x)) * sin(3*x), x = 0..Pi):
> " = value(");
```

$$\int_0^\pi \sin(z\sin(x))\sin(3\,x)\,dx =$$
$$8\,\frac{\pi\,\text{BesselJ}(1,\,z)}{z^2} - 4\,\frac{\pi\,\text{BesselJ}(0,\,z)}{z} \quad \text{sselJ}(1,\,z)$$

Az első-, másod- és harmadfajú elliptikus integrálok mind algebrai formában (a Jacobi-féle jelöléssel), mind a Legendre-féle alakban rendelkezésünkre állnak. Két jelölésmód létezik: az egyik a Byrd és Friedman [29], „Handbook of Elliptic Integrals" című könyvében használtat követi, a másik az [1]-nek („Handbook of Mathematical Functions") megfelelő. A részleteket lásd a 10.1. táblázatban.

Elnevezés	**Definíció**	**Maple függvény**
elsőfajú nem teljes elliptikus integrál	$\int_0^x \frac{1}{\sqrt{(1-t^2)(1-k^2t^2)}}\,dt$	LegendreF(x, k) EllipticF(x, k)
másodfajú nem teljes elliptikus integrál	$\int_0^x \frac{\sqrt{1-k^2t^2}}{\sqrt{1-t^2}}\,dt$	LegendreE(x, k) EllipticE(x, k)
harmadfajú nem teljes elliptikus integrál	$\int_0^x \frac{1}{(1-at^2)\sqrt{(1-t^2)(1-k^2t^2)}}\,dt$	LegendrePi(x, a, k) EllipticPi(x, a, k)
elsőfajú teljes elliptikus integrál	$\int_0^1 \frac{1}{\sqrt{(1-t^2)(1-k^2t^2)}}\,dt$	LegendreKc(k) EllipticK(k)
másodfajú teljes elliptikus integrál	$\int_0^1 \frac{\sqrt{1-k^2t^2}}{\sqrt{1-t^2}}\,dt$	LegendreEc(k) EllipticE(k)
harmadfajú teljes elliptikus integrál	$\int_0^1 \frac{1}{(1-at^2)\sqrt{(1-t^2)(1-k^2t^2)}}\,dt$	LegendrePic(a, k) EllipticPi(a, k)
asszociált elsőfajú teljes elliptikus integrál	$\int_0^1 \frac{1}{\sqrt{(1-t^2)(1-c^2t^2)}}\,dt$	LegendreKc1(k) EllipticCK(k)
asszociált másodfajú teljes elliptikus integrál	$\int_0^1 \frac{\sqrt{1-c^2t^2}}{\sqrt{1-t^2}}\,dt$	LegendreEc1(k) EllipticCE(k)
asszociált harmadfajú teljes elliptikus integrál	$\int_0^1 \frac{1}{(1-at^2)\sqrt{(1-t^2)(1-c^2t^2)}}\,dt$	LegendrePic1(a, k) EllipticCPi(a, k)

10.1. táblázat: Elliptikus integrálok a Maple rendszerben

Föltesszük, hogy $0 < k < 1$ és $c = \sqrt{1-k^2}$. A Maple legtöbbször az [1] szerinti jelöléseket preferálja. Néhány példa:

```
> Int( 1/sqrt((t^2-1)*(t^2-2)), t = 2..infinity ):
> " = value(");
```

$$\int_2^{\infty} \frac{1}{\sqrt{(t^2-1)\,(t^2-2)}}\,dt = \frac{1}{2}\sqrt{2}\,\mathrm{EllipticF}(\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2})$$

```
> readlib( radnormal ): # load simplification routine
> Int( 1/sqrt((t-1)*(t-2)*(t-3)*(t-4)),
>   t = 4..infinity ): " = radnormal( value(") );
```

$$\int_4^{\infty} \frac{1}{\sqrt{(t-1)(t-2)(t-3)(t-4)}}\,dt = 2(
-\mathrm{EllipticK}(-2\sqrt{2}\,3^{3/4}+4\sqrt{2}\,3^{1/4})
+\mathrm{EllipticF}(\frac{1}{4}\sqrt{2}+\frac{1}{4}\sqrt{3}\sqrt{2}, -2\sqrt{2}\,3^{3/4}+4\sqrt{2}\,3^{1/4}))(\sqrt{3}-2)$$

```
> Int( 1 / sqrt(t*(1-t)*(1+t)),
>   t = 0..1 ):  " = radnormal( value(") );
```

$$\int_0^1 \frac{1}{\sqrt{t\,(1-t)\,(1+t)}}\,dt = -4\,\frac{\mathrm{EllipticK}(3-2\sqrt{2})\,(3\sqrt{2}-4)}{\sqrt{2}-2}$$

```
> Int( 1 / ( 1 - 1/9*sin(t)^2 )^(5/2),
>   t = 0..Pi/2 ):  " = value(");
```

$$\int_0^{1/2\,\pi} \frac{1}{(1-\frac{1}{9}\sin(t)^2)^{5/2}}\,dt = -\frac{3}{8}\,\mathrm{EllipticK}(\frac{1}{3}) + \frac{17}{12}\,\mathrm{EllipticPi}(\frac{1}{9}, \frac{1}{3})$$

10.3. Numerikus integrálás

A számítógépes algebrai rendszerek általában pontos eredmények elérésére törekednek, de a Maple-ben van több numerikus számítási lehetőség, például numerikus integrálás is:

```
> int( exp( arcsin(x) ), x = 0..1 );
```

$$\int_0^1 e^{\arcsin(x)}\,dx$$

```
> evalf(");
```

$$1.905238690$$

```
> Int( exp(-2*t) * t * ln(t), t = 0..infinity ):
> [ ", value(") ];
```

$$\left[\int_0^{\infty} e^{(-2\,t)}\,t\ln(t)\,dt, -\frac{1}{4}\ln(2)+\frac{1}{4}-\frac{1}{4}\gamma\right]$$

```
> evalf(");
```

$$[-.06759071137, -.0675907114]$$

```
> evalf( "", 20 );
```

$$[-.067590711365369542506, -.067590711365369542 51]$$

Figyeljük meg, hogy az előbb az **Int** tétlen formát használtuk az „aktív" **int** helyett. Így elkerültük a pontos eredmény kiszámítását, s azonnal a numerikus közelítésre tértünk át az **evalf@Int** kompozícióval (még pontosabban az **evalf/int** eljárással). Az alapföltételezés szerinti integrálási módszer a Clenshaw–Curtis kvadratúra, de ha lassú a konvergencia (a szinguláris helyek közelsége miatt), a rendszer megpróbálja kiküszöbölni a szingularitásokat, vagy átvált egy adaptív, kettős-exponenciális kvadratúra módszerre. Egy adaptív Newton–Cotes módszer is elérhető, ha kisebb pontosság (például `Digits <= 15`) is elegendő. Az **evalf/int** hívás opcionális negyedik argumentuma a preferált integrációs módszert jelöli ki:

```
> evalf( Int( 1/sqrt(x), x = 0..1, 10, _Dexp ) );
```

$$1.999998825$$

A Maple **evalf/int** eljárása szingularitásokkal rendelkező analitikus integrandus kezelésére egyéb technikák mellett általánosított sorfejtéseket és változótranszformációkat használ. Az érdeklődő Olvasó forduljon [71, 75]-höz.

10.4. Integráltranszformációk

Ebben az alfejezetben a Laplace-, Fourier-, Mellin-, Hilbert- és Hankel-transzformációra mutatunk példákat. Az f függvény K magfüggvényhez tartozó $\mathcal{T}(f)$ integráltranszformáltjának általános definíciója a

$$\mathcal{T}(f)(s) = \int_a^b f(t)K(s,t)\,dt$$

képlettel adható meg, föltéve, hogy az integrál létezik. A Fourier-, Laplace-, és a Mellin-transzformáltak a legismertebbek, ezekhez rendre az e^{-ist}, e^{-st} és a t^{s-1} magfüggvény tartozik. A 10.2. táblázatban fölsoroljuk a Maple `inttrans` csomagjában megtalálható integráltranszformációkat. A polinomok racionális kifejezéseinek összegeként fölírható kifejezések és bizonyos más függvények (így a **Dirac**-, **Heaviside**- és a Bessel-függvények) Laplace-transzformáltja a Maple **laplace** eljárásával állítható elő. A megfelelő inverz Laplace-transzformáltat az **invlaplace** adja meg:

```
> with( inttrans );
```

[*addtable*, *fourier*, *fouriercos*, *fouriersin*, *hankel*, *hilbert*, *invfourier*, *invhilbert*, *invlaplace*, *laplace*, *mellin*]

```
> t * BesselJ(0,a*t);
```

$$t\,\mathrm{BesselJ}(0,\,a\,t)$$

```
> laplace( ", t, s );
```

$$\frac{s}{(s^2+a^2)^{3/2}}$$

Transzformáció	Definíció	Maple függvény
Fourier	$\int_{-\infty}^{\infty} f(t)e^{-ist}\,dt$	fourier$(f(t), t, s)$
Fourier–Bessel, Hankel	$\int_{0}^{\infty} f(t)tJ_n(st)\,dt$	hankel$(f(t), t, s, n)$
Fourier-koszinusz	$\sqrt{\frac{2}{\pi}}\int_{0}^{\infty} f(t)\cos(st)\,dt$	fouriercos$(f(t), t, s)$
Fourier-szinusz	$\sqrt{\frac{2}{\pi}}\int_{0}^{\infty} f(t)\sin(st)\,dt$	fouriersin$(f(t), t, s)$
Hilbert	$\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{f(t)}{t-s}\,dt$	hilbert$(f(t), t, s)$
Laplace	$\int_{0}^{\infty} f(t)e^{st}\,dt$	laplace$(f(t), t, s)$
Mellin	$\int_{0}^{\infty} f(t)t^{s-1}\,dt$	mellin$(f(t), t, s)$

10.2. táblázat: Integráltranszformációk a Maple rendszerben

```
> invlaplace( ", s, t );
```

$$t\,\mathrm{BesselJ}(0,\, a\,t)$$

```
> (cosh(3*t) - 3*t*sinh(3*t) - 1) / t^2;
```

$$\frac{\cosh(3\,t) - 3\,t\sinh(3\,t) - 1}{t^2}$$

```
> laplace( ", t, s );
```

$$-s\ln(s) + \frac{1}{2}\ln(s-3)\,s + \frac{1}{2}\ln(3+s)\,s$$

```
> invlaplace( ", s, t );
```

$$-\mathrm{invlaplace}(s\ln(s),\, s,\, t) + \frac{1}{2}\,\mathrm{invlaplace}(\ln(s-3)\,s,\, s,\, t)$$
$$+\,\frac{1}{2}\,\mathrm{invlaplace}(\ln(3+s)\,s,\, s,\, t)$$

Túl optimisták voltunk: a Maple nem tudta belső táblázataiban tárolt inverz Laplace-transzformációkká konvertálni a feladatot. Kicsit segítenünk kell. Először is irjuk át a formulát és tegyük föl, hogy $s > 3$:

```
> factor( "" );
```

$$\frac{1}{2}\, s\, (-2\ln(s) + \ln(s-3) + \ln(3+s))$$

```
> assume( s>3 );
> combine( ", ln, integer );
```

$$\frac{1}{2}\, s\tilde{}\, \ln\left(\frac{(s\tilde{} - 3)\,(3 + s\tilde{})}{s\tilde{}^{2}}\right)$$

Most már ki tudja számolni a Maple az inverz Laplace-transzformáltat:

```
> invlaplace( ", s, t );
```

$$-\frac{3}{2}\,\frac{e^{(3\,t)}}{t} + \frac{3}{2}\,\frac{e^{(-3\,t)}}{t} - \frac{1}{t^2} + \frac{1}{2}\,\frac{e^{(3\,t)}}{t^2} + \frac{1}{2}\,\frac{e^{(-3\,t)}}{t^2}$$

Ellenőrizzük az eredményt:

```
> convert( ", 'trig' );
```

$$-\frac{3}{2}\,\frac{\cosh(3\,t) + \sinh(3\,t)}{t} + \frac{3}{2}\,\frac{\cosh(3\,t) - \sinh(3\,t)}{t} - \frac{1}{t^2}$$
$$+ \frac{1}{2}\,\frac{\cosh(3\,t) + \sinh(3\,t)}{t^2} + \frac{1}{2}\,\frac{\cosh(3\,t) - \sinh(3\,t)}{t^2}$$

```
> normal(");
```

$$-\frac{-\cosh(3\,t) + 3\,t\sinh(3\,t) + 1}{t^2}$$

```
> combine("); # get rid of the minus sign (if necessary)
```

$$\frac{\cosh(3\,t) - 3\,t\sinh(3\,t) - 1}{t^2}$$

A következő példa azt mutatja, hogy integráltranszformációk alkalmazásával bővíthetjük a Maple által megoldható integrálási feladatok osztályát:

```
> integrate( x^5*BesselJ(0,x), x = 0..t );
```

$$\int_0^t x^5\, \mathrm{BesselJ}(0, x)\, dx$$

Belátható azonban, hogy az

$$\int_0^t x^m J_n(x)\, dx$$

integrál kifejezhető a Bessel-függvények és t hatványai segítségével, ha $m + n$ páratlan. Segítsünk a Maple-nek a Laplace-transzformált kiszámításában, a részeredmények egyszerűsítésében, az inverz Laplace-transzformációban és az integrációs konstans vizsgálatában:

```
> laplace( ", t, s );
```

$$\frac{945\,\dfrac{s^5}{(s^2+1)^{11/2}} - 1050\,\dfrac{s^3}{(s^2+1)^{9/2}} + 225\,\dfrac{s}{(s^2+1)^{7/2}}}{s}$$

```
> normal(");
```

$$15\,\frac{8\,s^4-40\,s^2+15}{(s^2+1)^{11/2}}$$

```
> invlaplace( ", s, t );
```

$$\frac{8}{21}\,t^3\,\mathrm{BesselJ}(3,\,t)+\frac{16}{21}\,t^4\,\mathrm{BesselJ}(2,\,t)+\frac{8}{63}\,t^5\,\mathrm{BesselJ}(1,\,t)$$
$$-\frac{40}{63}\,t^4\,\mathrm{BesselJ}(4,\,t)-\frac{40}{63}\,t^5\,\mathrm{BesselJ}(3,\,t)+\frac{5}{21}\,t^5\,\mathrm{BesselJ}(5,\,t)$$

```
> eval( subs( t=0, " ) );
```

$$0$$

Ellenőrizzük differenciálással az eredményt:

```
> expand( diff( "", t ) );
```

$$t^5\,\mathrm{BesselJ}(0,\,t)$$

Tehát a következő konklúzióra jutottunk:

$$\int x^5\,J_0(x)\,dx =$$

$$\frac{8}{21}\,x^3\,J_3(x)+\frac{16}{21}\,x^4\,J_2(x)+\frac{8}{63}\,x^5\,J_1(x)-\frac{40}{63}\,x^4\,J_4(x)-\frac{40}{63}\,x^5\,J_3(x)+\frac{5}{21}\,x^5\,J_5(x)$$

A többi integráltranszformációhoz hasonlóan a Laplace-transzformáció legfontosabb alkalmazásai is a differenciál- és az integrálegyenletek területére esnek. A differenciálegyenleteket a 17. fejezetben fogjuk részletesen vizsgálni. Itt az integrálegyenletekre adunk egy példát:

```
> int_eqn := integrate( exp(a*x) * f(t-x), x = 0..t )
>   + b*f(t) = t;
```

$$\mathit{int_eqn} := \int_0^t e^{(a\,x)}\,\mathrm{f}(t-x)\,dx+b\,\mathrm{f}(t)=t$$

```
> laplace( ", t, s );
```

$$\frac{\mathrm{laplace}(\mathrm{f}(t),\,t,\,s)}{s-a}+b\,\mathrm{laplace}(\mathrm{f}(t),\,t,\,s)=\frac{1}{s^2}$$

```
> readlib(isolate)( ", laplace( f(t), t, s ) );
```

$$\mathrm{laplace}(\mathrm{f}(t),\,t,\,s)=\frac{1}{s^2\,(\frac{1}{s-a}+b)}$$

```
> invlaplace( ", s, t );
```

$$\mathrm{f}(t)=\frac{1}{(-1+b\,a)^2}+\frac{a\,t}{-1+b\,a}-\frac{e^{(\frac{(-1+b\,a)\,t}{b})}}{(-1+b\,a)^2}$$

```
> map( factor, " );
```

$$\mathrm{f}(t)=\frac{1-a\,t+a^2\,t\,b-e^{(\frac{(-1+b\,a)\,t}{b})}}{(-1+b\,a)^2}$$

Ellenőrizzük az eredményt.

```
> f := unapply( rhs("), t );
```

$$f := t \rightarrow \frac{1 - a\,t + a^2\,t\,b - e^{\left(\frac{(-1+b\,a)\,t}{b}\right)}}{(-1+b\,a)^2}$$

```
> testeq( int_eqn );
```

true

A **fourier** Maple eljárással polinomok racionális függvényeiből képezett összegek Fourier-transzformáltját határozhatjuk meg:

```
> 1/(1+t^3);
```

$$\frac{1}{1+t^3}$$

```
> fourier( ", t, omega );
```

$$\frac{1}{3}\,I\,e^{(I\,\omega)}\,\pi\,(\mathrm{Heaviside}(-\omega) - \mathrm{Heaviside}(\omega)) + e^{(-1/2\,I\,\omega)}\big(\\ \frac{1}{3}\,I\,(-1 - I\,\sqrt{3})\,e^{(1/2\,\sqrt{3}\,\omega)}\,\pi\,\mathrm{Heaviside}(-\omega)\\ -\frac{1}{3}\,I\,(-1 + I\,\sqrt{3})\,e^{(-1/2\,\sqrt{3}\,\omega)}\,\pi\,\mathrm{Heaviside}(\omega)\big)$$

A választ úgy ellenőrizhetjük, hogy kiszámítjuk az inverz Fourier-transzformáltat az **invfourier**-vel:

```
> invfourier( ", omega, t );
```

$$-\frac{4}{(t+1)\,(-I+\sqrt{3}+2\,I\,t)\,(-\sqrt{3}-I+2\,I\,t)}$$

```
> normal( ", 'expanded' );
```

$$\frac{1}{1+t^3}$$

A Maple konvolúciós módszereket, táblabeli kereséseket és határozott integrálást is tud alkalmazni az olyan speciális függvények kezelésére, mint a trigonometrikus függvények, a **Dirac**- és a **Heaviside**-függvények, az exponenciális függvény és a Bessel-függvények.

```
> fourier( BesselJ(0,t), t, omega );
```

$$-2\,\frac{\mathrm{Heaviside}(\omega-1) - \mathrm{Heaviside}(\omega+1)}{\sqrt{1-\omega^2}}$$

```
> fourier( BesselJ( 0, sqrt(t^2+1) ), t, omega );
```

$$2\,\frac{e^{(I\,\omega)}\cos(-1+\omega^2)\,(\mathrm{Heaviside}(\omega+1) - \mathrm{Heaviside}(\omega-1))}{\sqrt{1-\omega^2}}$$

Az **addtable** eljárással adhatjuk meg az általunk definiált függvények Fourier-transzformáltját. Például a Maple nem ismeri az $x/\sinh x$ transzformáltját. Ha ezt a függvényt elnevezzük mondjuk F-nek, akkor az alábbiak szerint tudjuk hozzáfűzni a **fourier** keresési táblázatához:

```
> addtable(
>   fourier, # name of integral transform
>   F(t), # name of user function
>   2/Pi*exp(Pi*w)/(1+exp(Pi*w))^2, # transform
>   t, w # variables used in transform
> );
> fourier( F(x), x, omega );
```

$$2\,\frac{e^{(\pi\,\omega)}}{\pi\,(1+e^{(\pi\,\omega)})^2}$$

```
> fourier( x^2 * F(x), x, omega );
```

$$-2\,\frac{\pi\,e^{(\pi\,\omega)}}{(1+e^{(\pi\,\omega)})^2}+12\,\frac{(e^{(\pi\,\omega)})^2\,\pi}{(1+e^{(\pi\,\omega)})^3}-12\,\frac{(e^{(\pi\,\omega)})^3\,\pi}{(1+e^{(\pi\,\omega)})^4}$$

```
> simplify(");
```

$$-2\,\frac{\pi\,e^{(\pi\,\omega)}\,(1-4\,e^{(\pi\,\omega)}+e^{(2\,\pi\,\omega)})}{(1+e^{(\pi\,\omega)})^4}$$

Ugyanígy bővíthető az **addtable** fölhasználásával a többi integráltranszformáció keresési táblázata is.

A **fourier** és az **invfourier** eljárás a szimbolikus Fourier-transzformáltak és inverzeik meghatározását végzi. Az `FFT` és az `iFFT` a numerikus gyors fourier transzformációt (**F**ast **F**ourier **T**ransform), illetve annak inverzét hajtja végre. Elevenítsük föl a komplex számokból álló N hosszúságú $x=[x_0,x_1,\ldots,x_{N-1}]$ lista $X=[X_0,X_1,\ldots,X_{N-1}]$ Fourier transzformáltjának definícióját. Tetszőleges $0\le k\le N-1$-re

$$X[k]=\sum_{j=0}^{N-1}x_j e^{\frac{-2\pi ijk}{N}}.$$

Ha N valamilyen kettő hatvánnyal egyenlő, a [47]-ben leírt gyors Fourier transzformáció alkalmazható. Az első példában $N=2^3$ és a valós számokból álló $[-1,-1,-1,-1,1,1,1,1]$ sorozat gyors Fourier transzformáltját számoljuk ki:

```
> readlib( FFT ): # load the procedure FFT
> x := array([-1,-1,-1,-1,1,1,1,1]): # real parts of data
> y := array([0,0,0,0,0,0,0,0]): # imaginary parts of data
> FFT(3,x,y): # transform data
> print(x); # real parts of transformed data
```

$$[0,\,-2.000000001,\,0,\,-1.999999999,\,0,\,-1.999999999,\,0,\,-2.000000001\,]$$

```
> print(y); # imaginary parts of transformed data
```

$$[0,\,4.828427122,\,0,\,.828427124,\,0,\,-.828427124,\,0,\,-4.828427122]$$

Ha a komplex számokat a hagyományosabb alakban akarjuk írni, hasznos lehet a következő **zip** eljárás:

```
> # normal komplex notation
> zip( (a,b) -> a+b*I, x, y ): convert( ", list );
```

$$[0, -2.000000001 + 4.828427122\,I, 0, -1.999999999 + .828427124\,I, 0, -1.999999999 - .828427124\,I, 0, -2.000000001 - 4.828427122\,I]$$

```
> iFFT(3,x,y): # check results
> print(x);
```

$$[-1.000000000, -.9999999990, -.9999999995, -.9999999985, 1.000000000, .9999999990, .9999999995, .9999999985]$$

```
> print(y);
```

$$[0, .2500000000\,10^{-9}, 0, -.2500000000\,10^{-9}, 0, -.2500000000\,10^{-9}, 0, .2500000000\,10^{-9}]$$

Az **fnormal** eljárás akkor használható, ha kis abszolút értékű számokat 0-ra akarunk normalizálni:

```
> y := map( fnormal, y );
```

$$y := [0, 0, 0, 0, 0, 0, 0, 0]$$

A gyors Fourier transzformáció gyakori és fontos alkalmazása konvolúciók végzése az adatok simítása céljából. Az alábbiakban erre találunk egy egyszerű, bár mesterkélt példát. A részleteket nem kommentáljuk, az itt használt minden adattípussal és eljárással foglalkozunk még a könyv hátralévő részében.

Először is betöltjük a stats csomag standard normális eloszlású véletlenszám-generátorát:

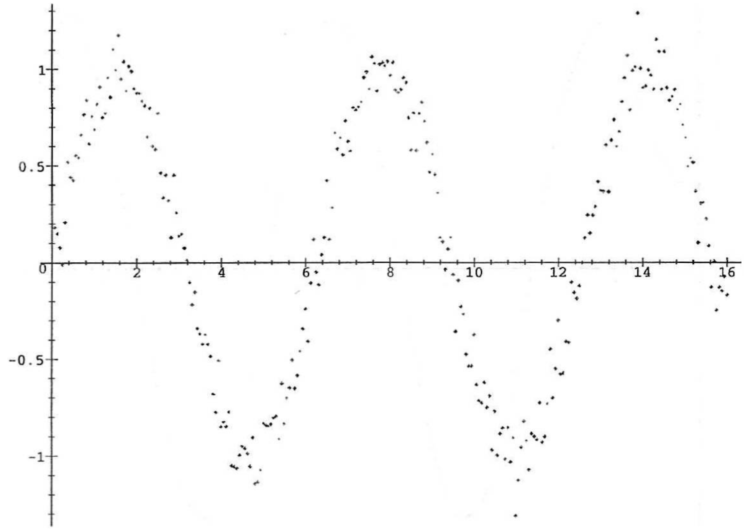
```
> noise := stats[ random, normald ]:
```

Ezután adatokat generálunk; az adatok valós és képzetes részét külön állítjuk elő:

```
> re_data := array( [ seq( sin(0.0625*k) + 0.1*noise(),
>   k=1..2^8) ] ):
> im_data := array( [ seq( 0, k=1..2^8) ] ):
```

Hogy legyen valami elképzelésünk a kapott adatokról, kirajzoltatjuk őket. Amint a 10.2. ábráról látható, a konstrukció miatt a szinusz függvényhez hasonló grafikont kapunk.

```
> xcoords := array( [ seq( 0.0625*k, k=1..2^8) ] ):
> plotdata := convert( zip( (a,b) -> [a,b], xcoords,
>   re_data ), list ):
> plot( plotdata, style = POINT );
```



10.2. ábra: Zajos szinusz függvény

Az adatok simítására a $t \to \exp(-\frac{100}{256}t^2)$ magfüggvényt használjuk.

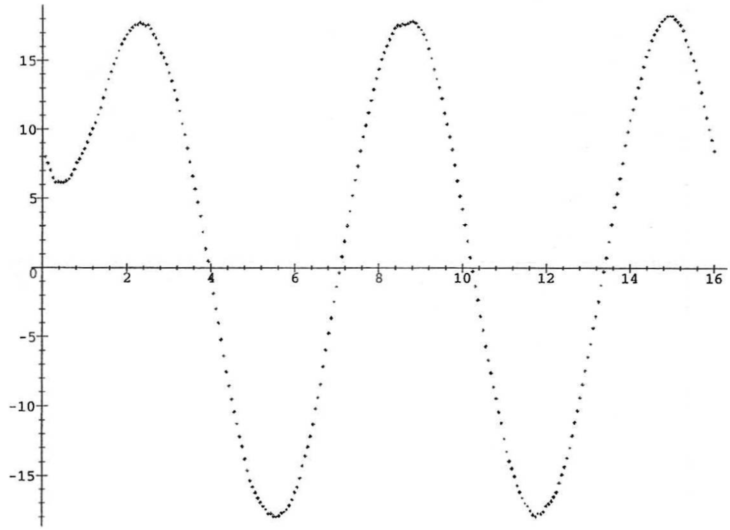
```
> re_kernel := array( [ seq( exp( -100.0 * (k/2^8)^2 ),
>   k=1..2^8 ) ] ):
> im_kernel := array( [ seq( 0, k=1..2^8) ] ):
```

Kiszámítjuk az adatoknak a magfüggvényre vonatkozó gyors Fourier transzformáltját, az eredményt komplex számokból álló tömbökként írjuk föl:

```
> FFT( 8, re_data, im_data ):
> FFT( 8, re_kernel, im_kernel ):
> data := zip( (a,b) -> (a+b*I), re_data, im_data ):
> kernel := zip( (a,b) -> (a+b*I), re_kernel, im_kernel ):
> newdata := zip( (a,b) -> a*b, data, kernel ):
> new_re_data := map( Re, newdata ):
> new_im_data := map( Im, newdata ):
```

Befejezésül inverze gyors Fourier transzformációt alkalmazunk az adatok Fourier transzformáltjának és a magfüggvénynek szorzatára. A végeredmény csak egy skalárral tér el az eredeti adatok simított változatától. Ez jól megfigyelhető a 10.3. ábrán.

```
> iFFT( 8, new_re_data, new_im_data ):
> plotdata := convert( zip( (a,b) -> [a,b], xcoords,
>   new_re_data), list ):
> plot( plotdata, style=POINT );
```



10.3. ábra: Adatsimítás

A fejezetet a Maple-ben rendelkezésünkre álló további integráltranszformációkra vonatkozó példákkal zárjuk:

- Hankel transzformáció

```
> with( inttrans ):
> assume( k, integer, k>0 ):
> hankel( sqrt(t), t, s, k );
```

$$2\,\frac{\sqrt{2}\,\Gamma(\frac{5}{4}+\frac{1}{2}\,k\tilde{})}{s^{5/2}\,\Gamma(\frac{1}{2}\,k\tilde{}-\frac{1}{4})}$$

- Fourier szinusz és Fourier koszinusz transzformáció

```
> assume( a>0 ):
> fouriercos( Heaviside(a-t), t, s );
```

$$\frac{\sqrt{2}\sin(a\tilde{}\,s)}{\sqrt{\pi}\,s}$$

```
> fouriercos( ", s, t );
```

$$\frac{1}{2}-\frac{1}{2}\,\mathrm{signum}(t-a\tilde{})$$

```
> convert( ", Heaviside );
```

$$1-\mathrm{Heaviside}(t-a\tilde{})$$

- Hilbert transzformáció

```
> assume( k, integer, k>0 ):
> hilbert( Dirac(x) + sin(k*x)/x, x, y );
```

$$\frac{-1 + \pi \cos(k\tilde{}\, y) - \pi}{y\,\pi}$$

- Mellin transzformáció

```
> mellin( 1 / (1+t), t, s );
```

$$\frac{\pi}{\sin(\pi\, s)}$$

```
> mellin( ln(1+t), t, s );
```

$$-\frac{\pi}{\sin(\pi\,(s+1))\, s}$$

10.5. Hogyan segítsünk a Maple-nek az integrálásnál?

Ebben a részben néhány olyan integrálási problémát vizsgálunk meg, amelyre a Maple magára hagyva nem képes megoldást találni, de némi segítséggel meg tudja oldani a feladatot. Az emberi segítségre elég gyakran szükség van, ha a feladatban nem elemi függvények is szerepelnek, vagy ha a paramétereknek bizonyos föltételeket kell kielégíteni.

Az első integrálási probléma paraméterének ki kell elégítenie bizonyos föltételt ahhoz, hogy az integrált analitikusan meg lehessen határozni.

```
> Int( exp(-c*x^2), x = 0..infinity ):  " = value(");

Definite integration:
Can't determine if the integral is convergent.
Need to know the sign of --> c
Will now try indefinite integration and then take limits.
```

$$\int_0^{\infty} e^{(-c\,x^2)}\, dx = \lim_{x\to\infty} \frac{1}{2}\,\frac{\sqrt{\pi}\,\mathrm{erf}(\sqrt{c}\,x)}{\sqrt{c}}$$

Gyakori logikai hiba, hogy *mi* ugyan pozitív valós konstansnak tekintjük c-t, de a *Maple* ettől még nem fog ugyanebből kiindulni. A fenti válaszból kitűnik, hogy a Maple legalábbis a határozatlan integrált ismeri. Ahhoz, hogy a határozott integrál kiszámításában előbbre léphessünk, az **assume**-mal fölkell tennünk, hogy c pozitív valós konstans; nem pozitív értékekre ugyanis az integrál divergens.

```
> assume( c>0):
> Int( exp(-c*x^2), x = 0..infinity ):  " = value(");
```

$$\int_0^{\infty} e^{(-c\tilde{}\, x^2)}\, dx = \frac{1}{2}\,\frac{\sqrt{\pi}}{\sqrt{c\tilde{}}}$$

Az alábbi integrálási példában a Maple elegendő információ hiányában nem kezd el automatikusan dolgozni:

```
> int( sqrt( (x^2-a^2) * (b^2-x^2) ), x = a..b );
```

$$\int_a^b \sqrt{(x^2-a^2)(b^2-x^2)}\,dx$$

Ilyesmi gyakran megesik: elfelejtettük a Maple-lel tudatni a paraméterekre vonatkozó föltételeket. Javítsuk ki ezt a hibát:

```
> assume( a>0, b>0 ):
> infolevel[int] := 2:
> int( sqrt( (x^2-a^2) * (b^2-x^2) ), x = a..b );

int/ellalg/trxlgdre:   cannot determine sign:  1/a~^2-1/b~^2
int/indef:   first-stage indefinite integration
int/algebraic/algebraic:   algebraic integration
int/rischnorm:   enter Risch-Norman integrator
int/risch:   enter Risch integration
int/risch:   enter Risch integration
int/risch:   the field extensions are
```

$$[_X,\ _root(-(_X^2-a^{\sim 2})(-b^{\sim 2}+_X^2),2)]$$

```
int/risch:   Introduce the namings:
```

$$\{_th_1=_root(-(_X^2-a^{\sim 2})(-b^{\sim 2}+_X^2),2)\}$$

```
unknown:   integrand is
```

$$\frac{(-_X^2+a^{\sim 2})(b^{\sim 2}-_X^2)}{_th_1}$$

```
int/risch/alg1:   integrand is
```

$$\frac{(-_X^2+a^{\sim 2})(b^{\sim 2}-_X^2)}{_th_1}$$

```
int/risch/alg1:   integral expressed as
```

$$\frac{1}{3}\frac{_X(-_X^2+a^{\sim 2})(b^{\sim 2}-_X^2)}{_th_1} + \int \frac{\frac{2}{3}a^{\sim 2}b^{\sim 2}-\frac{1}{3}_X^2a^{\sim 2}-\frac{1}{3}_X^2b^{\sim 2}}{_th_1}\,d_X$$

```
int/indef:   first-stage indefinite integration
int/indef2:   second-stage indefinite integration
int/indef2:   trying integration by parts
int/risch:   exit Risch integration
int/def:   definite integration
int/def:   definite integration
int/contour:   contour integration
```

$$\int_{a^{\sim}}^{b^{\sim}} \sqrt{(x^2 - a^{\sim 2})(b^{\sim 2} - x^2)}\, dx$$

```
> infolevel[int] := 1:
```

Még mindig nem kaptunk választ, de szerencsére az `infolevel[int]` értékének 2-re növelésével a Maple sokkal közlékenyebbé vált. A képernyőn megjelenő rengeteg üzenetben a Maple elmondta, hogy éppen mit csinált. Már az első üzenetek egyike így szólt:

```
int/ellalg/trxlgdre:   cannot determine sign:   1/a~^2-1/b~^2
```

Itt a megoldás kulcsa: tegyük föl azt is, hogy $b > a$. Ezt *mi* bizonyára természetesnek vettük, nem így a *Maple*:

```
> additionally( b>a ):
> int( sqrt( (x^2-a^2) * (b^2-x^2) ), x = a..b );
```

$$-\frac{2}{3}\, a^{\sim 2}\, b^{\sim}\, \mathrm{EllipticK}(\sqrt{1 - \frac{a^{\sim 2}}{b^{\sim 2}}})$$
$$+ \frac{a^{\sim 2}\, (\frac{1}{3}\, a^{\sim 2} + \frac{1}{3}\, b^{\sim 2})\, \mathrm{EllipticPi}(1 - \frac{a^{\sim 2}}{b^{\sim 2}}, \sqrt{1 - \frac{a^{\sim 2}}{b^{\sim 2}}})}{b^{\sim}}$$

Még egy záró megjegyzés ehhez a példához. Ha valami okból azt sejtjük, hogy elliptikus integrálok is föllépnek, az `infolevel[elliptic]` változónak nagyobb értéket adva is informálódhatunk a kifejezések előjeléről anélkül, hogy a Maple számos egyéb próbálkozását követnünk kellene.

Meglehetősen bonyolult feladat a megfelelő föltételek kiválasztása, melyekkel legtöbbet segíthetünk a Maple-nek az integrálási feladatok megoldásában. Ehhez gyakran szükség van minden matematikai ismeretünkre és számítási tapasztalatunkra. A [141]-ben leírt módon úgy is átdefiniálhatjuk a beépített **signum** eljárást, hogy számolás közben a Maple információkat kérjen az előjelekről. A módszer hátránya, hogy a Maple olyan szituációkban is kérhet információkat, amelyekről nem világos, hogy mi közük van a megoldandó problémához.

Néha utalnunk kell a követendő módszerre, például a változó helyettesítésére vagy a parciális integrálásra. A Maple `student` csomagjával ezt kényelmesen megtehetjük:

```
> with( student ):  # load the student package
> Int( sqrt( x + sqrt(x) ), x );
```

$$\int \sqrt{x + \sqrt{x}}\, dx$$

```
> changevar( sqrt(x)=y, ", y );
```

$$\int 2\sqrt{y^2 + y}\, y\, dy$$

```
> "" = value(");
```

$$\int \sqrt{x+\sqrt{x}}\,dx =$$
$$\frac{2}{3}(y^2+y)^{3/2} - \frac{1}{4}(2y+1)\sqrt{y^2+y} + \frac{1}{8}\ln(y+\frac{1}{2}+\sqrt{y^2+y})$$

```
> subs( y=sqrt(x), " );
```

$$\int \sqrt{x+\sqrt{x}}\,dx =$$
$$\frac{2}{3}(x+\sqrt{x})^{3/2} - \frac{1}{4}(2\sqrt{x}+1)\sqrt{x+\sqrt{x}} + \frac{1}{8}\ln(\sqrt{x}+\frac{1}{2}+\sqrt{x+\sqrt{x}})$$

```
> radnormal( diff(",x) ); # check answer
```

$$\sqrt{x+\sqrt{x}} = \sqrt{x+\sqrt{x}}$$

Egy másik példa:

```
> Int( x*exp(-a^2*x^2)*erf(b*x), x );
```

$$\int x\,e^{(-a^2 x^2)}\,\mathrm{erf}(b\,x)\,dx$$

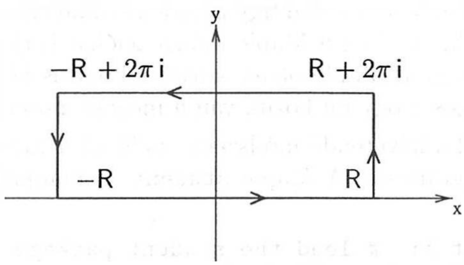
```
> intparts( ", erf(b*x) ): "" = value(");
```

$$\int x\,e^{(-a^2 x^2)}\,\mathrm{erf}(b\,x)\,dx = -\frac{1}{2}\,\frac{\mathrm{erf}(b\,x)\,e^{(-a^2 x^2)}}{a^2} + \frac{1}{2}\,\frac{\mathrm{erf}(\sqrt{b^2+a^2}\,x)\,b}{\sqrt{b^2+a^2}\,a^2}$$

Néha a vonalmenti integrálás jól ismert módszerét kell alkalmaznunk, s a Maple-t ehhez vetjük be számolási segédeszközként. Az alábbiakban ezen a módon határozzuk meg

$$\int_{-\infty}^{\infty} \frac{e^{ax}}{1+e^{ax}}\,dx$$

értékét a $0 < a < 1$ föltétel mellett. Először az x valós változót a z komplex változóval helyettesítjük, s a 10.4. ábrán látható vonal mentén integrálunk.



10.4. ábra: Az integrálás görbéje

Ha az $R \to \infty$ határértékét vesszük, a görbe vízszintes részei a keresett integrált adják, a függőleges részekhez tartozó integrál pedig eltűnik az a-ra tett föltétel miatt:

$$\oint_C \frac{e^{az}}{1+e^{az}}\,dz = (1-e^{2\pi a})\lim_{R\to\infty}\int_{-\infty}^{\infty} \frac{e^{ax}}{1+e^{ax}}\,dx$$

A vonalintegál értéke $2\pi i \sum$ *„a téglalapba eső reziduumok összege"*. Keressük meg tehát a nevező itteni zérushelyeit és számítsuk ki a hozzájuk tartozó reziduumokat.

```
> exp(a*z) / ( 1 + exp(z) );
```

$$\frac{e^{(a\,z)}}{1+e^z}$$

```
> solve( denom("), z ); # the solution in the strip
```

$$I\,\pi$$

```
> readlib( residue )( "", z = " );
```

$$-e^{(I\,a\,\pi)}$$

```
> ( 1 - exp(2*Pi*a*I) ) * integral = 2*Pi*I * ";
```

$$(1-e^{(2\,I\,a\,\pi)})\,\mathit{integral} = -2\,I\,\pi\,e^{(I\,a\,\pi)}$$

```
> (lhs/rhs)(") = 1;
```

$$\frac{1}{2}\,\frac{I\,(1-e^{(2\,I\,a\,\pi)})\,\mathit{integral}}{\pi\,e^{(I\,a\,\pi)}} = 1$$

```
> simplify( ", exp );
```

$$\frac{\mathit{integral}\,\sin(a\,\pi)}{\pi} = 1$$

```
> Int( exp(a*x)/(1+exp(x)), x = -infinity..infinity )
>   = solve( ", integral );
```

$$\int_{-\infty}^{\infty} \frac{e^{(a\,x)}}{1+e^x}\,dx = \frac{\pi}{\sin(a\,\pi)}$$

Az olyan szokásos trükkök, mint az integrálás előtti, paraméter szerinti differenciálás könnyen végrehajthatók a Maple-ben. Egy példa:

```
> assume( a>0 ):
> int( tanh(a*x/2) / (x*cosh(a*x)) , x = 0..infinity );
```

$$\int_0^{\infty} \frac{\tanh(\frac{1}{2}\,a\tilde{}\,x)}{x\cosh(a\tilde{}\,x)}\,dx$$

```
> diff( ", a );
```

$$0$$

A határozott integrál értéke tehát nem függ az a pozitív paramétertől.

Végül, de nem utolsósorban egy jó tanács: ha a Maple-lel dolgozunk, használjuk agyunkat is, és kétszer is gondoljuk meg, mielőtt belekezdünk a számolásba. Például a Maple nem tudja meghatározni az

$$\int_{-\pi}^{\pi} \frac{\sin t}{1+\sin^8 t}\,dt$$

integrált, de erre nincs is szükség, mivel rögtön látható, hogy az integrandus páratlan függvény, s emiatt az integrál 0.

10.6. Összegzés

Számokból álló véges összegek könnyedén kiszámíthatók az **add** eljárással.

```
> Sum( k^7, k = 1..20 ) = add( k^7, k = 1..20 );
```

$$\sum_{k=1}^{20} k^7 = 3877286700$$

```
> primes := [11,31,41,71,101,131,181,191,211,241]:
> Sum( i, i = primes) = add( i, i = primes );
```

$$\sum_{i=\%1} i = 1210$$

$$\%1 := [11, 31, 41, 71, 101, 131, 181, 191, 211, 241]$$

```
> poly := add( a[i]*x^i, i=0..5 );
```

$$poly := a_0 + a_1\, x + a_2\, x^2 + a_3\, x^3 + a_4\, x^4 + a_5\, x^5$$

Lehetőség van „szimbolikus összegzésre" is. Először tekintsük a határozatlan összegzést: adott az

$$a_1, a_2, a_3, \ldots$$

sorozat, vagy pontosabban a sorozatot előállító kifejezés, és olyan s_k kifejezést szeretnénk találni, amelyben nem fordul elő az összegzés jele, továbbá

$$a_k = s_k - s_{k-1}.$$

Ez a feladat a határozatlan integrálás diszkrét analógja. A Maple a következő módszereket használja s_k megtalálására:

- A polinomok a Bernoulli polinomokon alapuló következő formulával összegezhetők:

  $$\sum_{k=0}^{n-1} k^m = \frac{1}{m+1} \sum_{k=0}^{m} \binom{m+1}{k} B_k n^{m+1-k},$$

  ahol a B_k Bernoulli számokat a

  $$B_0 = 1 \quad \text{és} \quad \sum_{k=0}^{m} \binom{m+1}{k} B_k = 0 \quad \text{ha } m \geq 0$$

  implicit rekurzióval definiáljuk.

- Moenck módszerével (lásd [140]) összegezhetők az összegzési index racionális függvényei. Az eredmény olyan összeg, amely egy racionális függvényből, a Psi Polygamma függvényből, továbbá ennek deriváltjaiból áll.

- Gosper [82] döntési eljárása a Risch-algoritmus diszkrét megfelelője. Hipergeometrikus s_k összegekre alkalmazható, ilyenkor $s_k/s_{k-1} \in \mathbb{Q}(k)$.

- A [120] kiterjesztett Gosper algoritmus a következő kérdéssel foglalkozik: adott nemnegatív m-re és adott a_k sorozatra keressünk az $a_k = s_k - s_{k-m}$ föltételt teljesítő s_k sorozatot, ha tudjuk, hogy s_k k-szoros hipergeometrikus összeg.

A Maple tartalmazza a [193, 194]-nek megfelelő, hipergeometrikus sorokat összegző és hipergeometrikus azonosságokat kezelő (kibővített) Wilf–Zeilberger módszer implementációját is. Az érdeklődő Olvasónak ajánljuk [120, 153, 177]-et, és a bennük található hivatkozásokat. A Maple `sumtools` csomagja határozatlan és határozott összegek explicit kezelését végzi. Sok esetben egyedül a **sum** eljárás is automatikusan megoldja a problémát. Néhány példa:

- A Bernoulli polinomok módszere

```
> Sum( k^7, k = 1..n ):  " = value(");
```

$$\sum_{k=1}^{n} k^7 =$$
$$\frac{1}{8}(n+1)^8 - \frac{1}{2}(n+1)^7 + \frac{7}{12}(n+1)^6 - \frac{7}{24}(n+1)^4 + \frac{1}{12}(n+1)^2$$

```
> factor(");
```

$$\sum_{k=1}^{n} k^7 =$$
$$\frac{1}{24} n^2 (3n^4 + 6n^3 - n^2 - 4n + 2)(n+1)^2$$

- Moenck módszere

```
> Sum( 1/(k^2+k)^3, k = 1..n ):  " = value(");
```

$$\sum_{k=1}^{n} \frac{1}{(k^2+k)^3} =$$
$$-\frac{-2 - 3n + 6(n+1)^2}{(n+1)^3} + 6\,\Psi(1, n+2) + 10 - \pi^2$$

```
> limit( rhs("), n = infinity );
```

$$10 - \pi^2$$

```
> Sum( 1/(k^5+2*k+1),k = 0..n ):  " = value(");
```

$$\sum_{k=0}^{n} \frac{1}{k^5+2\,k+1} = \left(\sum_{_\alpha=\%1} \left(\frac{4096}{11317} + \frac{2560}{11317}_\alpha^4 - \frac{1600}{11317}_\alpha^3 + \frac{1000}{11317}_\alpha^2 - \frac{625}{11317}_\alpha\right) \Psi(n+1-_\alpha)\right) - \left(\sum_{_\alpha=\%1} \left(\frac{4096}{11317} + \frac{2560}{11317}_\alpha^4 - \frac{1600}{11317}_\alpha^3 + \frac{1000}{11317}_\alpha^2 - \frac{625}{11317}_\alpha\right) \Psi(-_\alpha)\right)$$

$$\%1 := \mathrm{RootOf}(_Z^5 + 2\,_Z + 1)$$

```
> limit( rhs("), n = infinity );
```

$$-\left(\sum_{_\alpha=\%1} \left(\frac{1}{11317} (4096 + 2560\,_\alpha^4 - 1600\,_\alpha^3 + 1000\,_\alpha^2 - 625\,_\alpha)\,\Psi(-_\alpha)\right)\right)$$

$$\%1 := \mathrm{RootOf}(_Z^5 + 2\,_Z + 1)$$

```
> evalf(");
```

$$1.282579632$$

```
> Sum( 1/(k^2 - 4) , k = 3..infinity ):  " = value(");
```

$$\sum_{k=3}^{\infty} \frac{1}{k^2-4} = \frac{25}{48}$$

```
> Sum( 1/(3*k+1)/(3*k+2)/(3*k+3)/(3*k+4),
>   k = 0..infinity ):  " = value(");
```

$$\sum_{k=0}^{\infty} \frac{1}{(3\,k+1)\,(3\,k+2)\,(3\,k+3)\,(3\,k+4)} = \frac{1}{6} + \frac{1}{36}\pi\sqrt{3} - \frac{1}{4}\ln(3)$$

- Gosper módszere

```
> # see (SIAM Review, 1994, Problem 94-2)
> Sum( (-1)^(k+1)*(4*k+1)*(2*k)! /
> (k!*4^k*(2*k-1)*(k+1)!), k = 1..n ):
> " = value(");
```

$$\sum_{k=1}^{n} \frac{(-1)^{(k+1)}\,(4\,k+1)\,(2\,k)!}{k!\,4^k\,(2\,k-1)\,(k+1)!} = -2\,\frac{(n+2)\,(-1)^{(n+2)}\,(2\,n+2)!}{(n+1)!\,4^{(n+1)}\,(2\,n+1)\,(n+2)!} + 1$$

- Hipergeometrikus azonosságok

```
> Sum( (-1)^(n-k) * binomial(2*n,k)^2, k = 0..2*n ):
> " = value(");
```

$$\sum_{k=0}^{2n} (-1)^{(n-k)} \operatorname{binomial}(2n, k)^2 = \frac{(-1)^n \sqrt{\pi}}{2^{(-2n)}\, \Gamma(n+1)\, \Gamma(\frac{1}{2} - n)}$$

Végtelen összegek numerikus közelítésének kiszámolásakor a Maple a Levin-féle u-transzformáltat használja a konvergencia gyorsítására (v.ö. [91, 192]), ezzel még divergens összegekhez is rendel valamilyen számértéket. Célszerű a kapott eredményt mindig ellenőrizni:

```
> Sum( 1/k^2, k=1..infinity );
```

$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

```
> evalf(");
```

$$1.644934067$$

```
> value(""); # the exact answer.
```

$$\frac{1}{6}\pi^2$$

```
> evalf( " - "" ); # comparison
```

$$.1\,10^{-8}$$

```
> Sum( 3^(-k), k=1..infinity );
```

$$\sum_{k=1}^{\infty} 3^{(-k)}$$

```
> evalf(");
```

$$.5000000000$$

```
> value("");
```

$$\frac{1}{2}$$

```
> Sum( 3^k, k=1..infinity );
```

$$\sum_{k=1}^{\infty} 3^k$$

```
> evalf(");
```

$$-1.500000000$$

```
> value("");
```

$$\infty$$

```
> Sum( 1/k^3, k=1..infinity );
```

$$\sum_{k=1}^{\infty} \frac{1}{k^3}$$

```
> evalf(");
```

$$1.202056903$$

```
> value("");
```

$$\zeta(3)$$

```
> evalf(");
```

$$1.202056903$$

```
> Sum( 1/k^(1/3), k=1..infinity );
```

$$\sum_{k=1}^{\infty} \frac{1}{k^{1/3}}$$

```
> evalf(");
```

$$-.9733602484$$

```
> value("");
```

$$\infty$$

Ennél a két példánál:

$$\sum_{k=1}^{\infty} 3^k$$

és

$$\sum_{k=1}^{\infty} \frac{1}{\sqrt[3]{k}}$$

meg kell vizsgálni, hogy a kapott numerikus közelítés megfelel-e a divergens sorok összegzésére általunk használt definíciónak. A Maple válaszai elég furcsának tűnnek (pozitív sorok összege negatív), de valójában nem is olyan rendkívüliek. Az utolsó eredmény például a Riemann-féle ζ függvény analitikus folytatásával igazolható (ennek alátámasztására közelítsük $\zeta(1/3)$-ot a Maple-ben).

10.7. Gyakorlatok

1. Számítsuk ki a következő határozatlan integrálokat, és ellenőrizzük az eredményeket differenciálással, valamint egyszerűsítéssel.

(a) $\displaystyle\int \sqrt{e^x - 1}\, dx$

(b) $\displaystyle\int \frac{x}{(2ax - x^2)^{3/2}}\, dx$

(c) $\displaystyle\int \sqrt{x^2 - a^2}\, dx$

(d) $\displaystyle\int \frac{1}{x\sqrt{1 + x^2}}\, dx$

(e) $\displaystyle\int \sec^3 x\, dx$

(f) $\displaystyle\int \frac{1}{1+\sin x+\cos x}\, dx$

(g) $\displaystyle\int \frac{\ln x}{x(x^2+1)^2}\, dx$

2. Határozzuk meg tetszőleges egész n-re $\int x^n e^x\, dx$-et, az eredményt ellenőrizzük különböző n értékekre.

3. Számítsuk ki a következő kettős integrálokat:

(a) $\displaystyle\int_0^1 \left(\int_0^1 \frac{x-y}{(x+y)^3}\, dy\right) dx$

(b) $\displaystyle\int_0^1 \left(\int_0^1 \frac{x-y}{(x+y)^3}\, dx\right) dy$

(c) Hasonlítsuk össze (a) és (b) eredményét. A Maple hibázott, vagy valami másról van szó?

4. Számítsuk ki a következő határozott integrálokat:

(a) $\displaystyle\int_1^{10} \frac{4x^4+4x^3-2x^2-10x+6}{x^5+7x^4+16x^3+10x^2}\, dx$

(b) $\displaystyle\int_0^{\frac{\pi}{2}} x^4 \sin x \cos x\, dx$

(c) $\displaystyle\int_{1/7}^{1/5} \frac{1}{x\sqrt{5x^2-6x+1}}\, dx$

(d) $\displaystyle\int_{-2}^{-1} \frac{1}{x}\, dx$

5. Számítsuk ki a következő határozott integrálokat:

(a) $\displaystyle\int_0^1 \frac{1}{\sqrt{1-x^2}}\, dx$

(b) $\displaystyle\int_0^1 x \arctan x\, dx$

(c) $\displaystyle\int_0^\infty e^{-ax}\cos^2(bx)\, dx$ pozitív valós a-ra

(d) $\displaystyle\int_0^\infty \frac{\sin x}{x}\, dx$

(e) $\displaystyle\int_0^\infty e^{-x} \ln x\, dx$

(f) $\displaystyle\int_0^\infty \frac{e^{-ax\ln x}}{\sqrt{x}}\, dx$ pozitív valós a-ra

(g) $\displaystyle\int_0^\infty \frac{e^{-\sqrt{t}}}{t^{1/4}(1-e^{-\sqrt{t}})}\, dx$

(h) $\displaystyle\int_1^\infty \frac{1}{\sqrt{x^4-1}}\, dx$

(i) $\displaystyle\int_0^{\frac{\pi}{2}} \sqrt{\cos x}\, dx$

(j) $\displaystyle\int_1^3 \frac{1}{\sqrt{x^4+4x^2+3}}\, dx$

(k) $\displaystyle\int_0^{\frac{\pi}{2}} \sqrt{\tan x}\, dx$

(l) $\displaystyle\int_0^\infty \frac{\sin^2 ax \sin bx}{x}\, dx$

(m) $\displaystyle\int_0^\infty \frac{1}{\cosh ax}\, dx$ pozitív valós a-ra

6. Legyen F az $F(T) := \displaystyle\int_1^T \frac{\exp(-u^2T)}{u}\, du$ integrállal definiált függvény.

(a) Értelmezzük a megfelelő F Maple függvényt, és adjuk meg $F(2)$ egy közelítő értékét.

(b) Állítsuk elő (D-vel) az F' deriváltat és számítsuk ki $F'(2)$ értékét.

7. Jelölje A az $\{(x,y) \in \mathbb{R}^2 \mid 1/2 \le xy \le 2, 1 \le x \le 3\}$ tartományt. Számítsuk ki az

$$\int\int_A \frac{exp(1/xy)}{y^2(x+1)^2}\, dx\, dy$$

integrált.

8. Ebben a feladatban a Risch-féle algoritmust követve bebizonyítjuk, hogy az

$$\int \frac{\ln^2(x-1)}{x}\, dx$$

integrál nem fejezhető ki elemi függvényekkel. Ha az integrál benne lenne az elemi függvények osztályában, akkor a Liouville-elvből a következő előállítás adódna:

$$B_3(x)\ln^3(x-1) + B_2(x)\ln^2(x-1) + B_1(x)\ln(x-1) + B_0(x),$$

ahol $B_3(x), B_2(x)$ és $B_1(x)$ racionális függvények, és csak $B_0(x)$ tartalmazhat új logaritmusos bővítéseket.

(a) Állítsuk elő $B_0(x), \ldots, B_3(x)$ differenciálegyenletét.

(b) Mutassuk meg, hogy $B_0(x)$ racionális konstans.

(c) Igazoljuk, hogy $B_2(x)$ differenciálegyenlete nem oldható meg a racionális függvények körében. Ez azt bizonyítja, hogy $\dfrac{\ln^2(x-1)}{x}$ integrálja nem fejezhető ki elemi függvényekkel.

9. A reziduumok módszerével mutassuk meg, hogy

$$\int_0^{2\pi} \frac{1}{a^2\cos^2\theta + b^2\sin^2\theta}\,\theta = \frac{2\pi}{ab}$$

ahol a és b nemnulla valós számok, továbbá $\left|\dfrac{b-a}{b+a}\right| < 1$. Hasonlítsuk össze eredményünket a Maple-től kapott válasszal.

10. Oldjuk meg az

$$\sin t = \int_0^t J_0(t-\theta) f(\theta)\, d\theta.$$

integrálegyenletet Laplace-transzformációval.

11. Oldjuk meg az $f(t) = 1 + \displaystyle\int_0^t (t-\theta) f(\theta)\, d\theta$ integrálegyenletet.

12. Számítsuk ki az $\displaystyle\int_0^\infty \frac{x}{1+x^2} \sin\omega^2 x\, dx$ integrált.

13. Számítsuk ki az $\displaystyle\int_0^{\frac{\pi}{4}} \frac{1}{(1+k^2\sin^3 t)^3\sqrt{1+k^2\sin^2 t}}\,dx$ integrált.

14. Számítsuk ki az $\displaystyle\int_0^{\infty} \frac{x^a-1}{\ln x}\,dx$ integrált nemnegatív a-ra. 15. Számítsuk ki az $\displaystyle\int_0^{\infty} \frac{\ln x}{(x+a)(x-1)}\,dx$ integrált pozitív a-ra.

16. Számítsuk ki az $\displaystyle\int \frac{\ln(x^2+1)}{x^2+1}\,dx$ integrált.

17. Számítsuk ki az $\displaystyle\int \frac{x^2}{\sqrt{x^6+1}}\,dx$ integrált.

18. Számítsuk ki az $\displaystyle\int \tan\big(\frac{1}{3}\arctan(x)\big)\,dx$ integrált.

19. Számítsuk ki az $\displaystyle\int \arcsin^2(x/a)\,dx$ integrált.

20. Számítsuk ki az $\displaystyle\int \frac{1}{\sqrt{a+\sqrt{x}+b}}\,dx$ integrált.

21. Számítsuk ki az alábbi végtelen összegeket:

(a) $\displaystyle\sum_{k=0}^{\infty} \frac{2k+3}{(k+1)(k+2)(k+3)}$

(b) $\displaystyle\sum_{k=1}^{\infty} \frac{k^2+k-1}{(k+2)!}$

(c) $\displaystyle\sum_{k=2}^{\infty} \frac{k}{(k-1)^2(k+1)^2}$

(d) $\displaystyle\frac{1}{4}+\sum_{k=1}^{\infty} \frac{3k+2}{k^3(k+1)(k+2)}$

(e) $\displaystyle\sum_{k=0}^{\infty} \frac{k^3+7k^2+4k+6}{k^2(k^2+2)(k^2+2k+3}$

22. Számítsuk ki a $\sum_{k=0}^{n} \binom{2n}{2k} (-3)^k$ összeget.

23. Számítsuk ki a $\sum_{k=0}^{n} \binom{2k}{k} \left(\frac{1}{2}\right)^{2k}$ összeget.

24. Számítsuk ki az első 32 prímszám szorzatát.

25. Számítsuk ki a $\prod_{k=2}^{\infty} 1 - 1/k^2$ szorzatot. (Útmutatás: határozzunk meg zárt képletet a szorzatra, ha k értéke 2 és n között változik, majd a **limit** alkalmazásával számítsuk ki a határértékét, ha n végtelenhez tart.)

26. Számítsuk ki $\sum_{k=1}^{n}(mk - 1)$-et, és hasonlítsuk össze az eredményt a [84]-ben található 0.112 formulával.

27. Számítsuk ki $\sum_{k=1}^{n-1} k^2 x^2$-et, és hasonlítsuk össze az eredményt a [84]-ben található 0.114 formulával.

11. Sorok, közelítések és határértékekek

Az analízis négy témakörét vizsgáljuk ebben a fejezetben: a csonkított sorfejtéseket, a formális hatványsorokat, az approximáció elméletét és a határérték-számítást. A példákból kitűnik, hogy a csonkított sorfejtések különböző típusai, így a Taylor sorok, a Laurent sorok, a Puisseux sorok és a Csebisev sorok mind rendelkezésünkre állnak Maple-ben. A Padé és a Csebisev–Padé sorfejtés a numerikus közelítéseket számoló numapprox Maple csomagban található. A megfelelő esetekben a sorfejtések határértékek kiszámítására is fölhasználhatók, v. ö. [49].

11.1. Csonkított sorfejtések

Egyváltozós függvények Taylor sorát könnyen és gyorsan meghatározhatjuk a Maple segítségével. Például a $\sin(\tan x) - \tan(\sin x)$ függvény $x = 0$ körüli 25-öd rendű sorfejtését pillanatok alatt kiszámíthatjuk:

```
> taylor( sin(tan(x)) - tan(sin(x)), x=0, 25 );
```

$$-\frac{1}{30}x^7 - \frac{29}{756}x^9 - \frac{1913}{75600}x^{11} - \frac{95}{7392}x^{13} - \frac{311148869}{54486432000}x^{15} - \frac{10193207}{4358914560}x^{17} - \frac{1664108363}{1905468364800}x^{19} - \frac{2097555460001}{7602818775552000}x^{21} - \frac{374694625074883}{6690480522485760000}x^{23} + O(x^{25})$$

A **taylor** eljárás hívásakor csupán a függvényt, a változót, a kifejtés helyét és a csonkítás rendjét kell specifikálnunk. Az $O(x^{25})$ szimbólum jelzi, hogy a Maple a sorfejtéseket egy speciális belső adatstruktúrában tárolja:

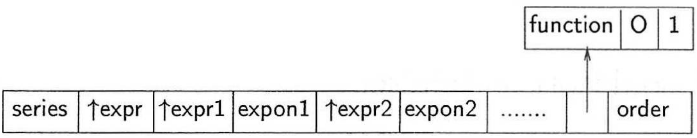
```
> whattype(");
```

series

```
> readlib(dismantle)("");

SERIES(22)
   NAME(4): x
   RATIONAL(3): -1/30
   [7]
   RATIONAL(3): -29/756
   [9]
   RATIONAL(3): -1913/75600
   [11]
   RATIONAL(3): -95/7392
   [13]
   RATIONAL(3): -311148869/54486432000
   [15]
   RATIONAL(3): -10193207/4358914560
   [17]
   RATIONAL(3): -1664108363/1905468364800
   [19]
   RATIONAL(3): -2097555460001/7602818775552000
   [21]
   RATIONAL(3): -374694625074883/6690480522485760000
   [23]
   FUNCTION(3)
      NAME(4): O
      EXPSEQ(2)
         INTPOS(2): 1
   [25]
```

A `series` adatstruktúra általános alakja a 11.1. ábrán látható.



11.1. ábra: A csonkított hatványsorok belső reprezentációja

Az ábrán szereplő `expr` kifejezés alakja általában `x-a`, ahol `x` az a változó, ami szerinti sorfejtést végeztünk, `a` pedig a sorfejtés helye. A további komponensek az együtthatókra és a kitevőkre mutató pointerpárok. Az adatvektor végén álló O(1) együtthatót a Maple a rendet meghatározó tagként interpretálja.

A fönti adattípussal egyezik meg a megfelelő **series** eljárás neve. Az eljárás alkalmazható általánosabb (például Laurant-féle) csonkított sorfejtések kiszámítására is:

```
> series( GAMMA(x), x=0, 2 );
```

$$x^{-1} - \gamma + (\frac{1}{12}\pi^2 + \frac{1}{2}\gamma^2)\,x + O(x^2)$$

A Taylor sorfejtés csonkításának rendje az O ordó szimbólumot tartalmazó tag megvizsgálásán kívül az **order** eljárással is megkapható:

```
> order(");
```

$$2$$

A **series** eljárás harmadik argumentuma elhagyható. A Maple ekkor az `Order` globális változó értéke alapján határozza meg a csonkítás rendjét. Ennek alapföltételezés szerinti értéke 6:

```
> Order;
```

$$6$$

```
> Order:= 3:  series( f(x), x=a );
```

$$f(a) + D(f)(a)\,(x-a) + \frac{1}{2}\,(D^{(2)})(f)(a)\,(x-a)^2 + O((x-a)^3)$$

```
> series( f(x)/(x-a)^2, x=a );
```

$$f(a)\,(x-a)^{-2} + D(f)(a)\,(x-a)^{-1} + \frac{1}{2}\,(D^{(2)})(f)(a) + O(x-a)$$

```
> series( 1/(cos(x)-sec(x)), x=0 );
```

$$-x^{-2} + O(x^{-1})$$

Az utolsó példa azt illusztrálja, hogy Laurant sorok esetében a „csonkítás rendje" a Maple által a sor meghatározása közben használt rendet jelenti. Lehet, hogy a végeredmény ennél kevesebb tagot tartalmaz. Ez a jelenség a numerikus számítások közben előforduló pontosságvesztésre hasonlít. De sorfejtések közben az ellenkező esettel is találkozhatunk:

```
> series( 1/(1-x^2), x=0, 5 );
```

$$1 + x^2 + x^4 + O(x^6)$$

Bár csak ötödrendű sorfejtést kértünk, a Maple egyedül eldöntötte, hogy az ötödrendű tag együtthatója 0, s tájékoztatott erről bennünket. A jelenséget sokszor az okozza, hogy a `remember` opció miatt a Maple korábbi, magasabb rendű sorfejtéseket is figyelembe vesz. Például a `series(cos(x), x=0, 25)` parancs végrehajtása után a koszinusz függvény minden sorfejtése legalább 25 tagot fog tartalmazni.

Ha csak Laurent sorokkal akarunk foglalkozni, a `numapprox` csomagból a **laurent** eljárást használjuk. Különben a Maple válaszában általánosabb, például Puisseux sorokat is megenged:

```
> 1/(x*(1+sqrt(x)));
```

$$\frac{1}{x\,(1+\sqrt{x})}$$

```
> series(",x);
```

$$\frac{1}{x}-\frac{1}{\sqrt{x}}+1-\sqrt{x}+x-x^{3/2}+x^2-x^{5/2}+x^3-x^{7/2}+x^4-x^{9/2}+x^5-x^{11/2}+\mathrm{O}(x^6)$$

A sorfejtésekben néha a Maple a független változót tartalmazó együtthatókat választ:

```
> series( x^(x^x), x=0, 5 );
```

$$x+\ln(x)^2\,x^2+(\frac{1}{2}\ln(x)^3+\frac{1}{2}\ln(x)^4)\,x^3+(\frac{1}{6}\ln(x)^4+\frac{1}{2}\ln(x)^5+\frac{1}{6}\ln(x)^6)\,x^4+\mathrm{O}(x^5)$$

```
> series( dilog(x), x=0, 3 );
```

$$\frac{1}{6}\pi^2+(\ln(x)-1)\,x+(\frac{1}{2}\ln(x)-\frac{1}{4})\,x^2+\mathrm{O}(x^3)$$

Az ilyen *általánosított sorfejtésekben* (v. ö. [72]) szereplő együttható-függvények növekedése legföljebb polinomiális lehet.

A Maple szép tulajdonsága, hogy integrállal vagy differenciálegyenlettel definiált függvények csonkított sorfejtését is képes kiszámolni. Nézzünk egy példát:

```
> integrate( ln(1+s*t)^2/(1+t^2), t=0..infinity );
```

$$\int_0^\infty \frac{\ln(1+s\,t)^2}{1+t^2}\,dt$$

```
> series( ", s=0, 5 );
```

$$\frac{1}{3}\pi^2\,s-\frac{1}{2}\pi\,s^2+(-\ln(s)+\frac{7}{6}-\frac{1}{9}\pi^2)\,s^3+\frac{11}{24}\pi\,s^4+\mathrm{O}(s^5)$$

A fönti sorfejtés kiszámítása három lépésben történik:

- a rendszer s szerint differenciálja és integrálja az integrandust,
- kiszámolja a kapott részeredmény $s=0$ körüli sorfejtését,
- integrálja a sorfejtés tagjait, és meghatározza a helyes integrációs állandót.

Másik példánk a kétdimenziós ingát leíró differenciálegyenlet megoldásának sorfejtése (lásd a 17.3. ábrát).

```
> DESol( l*diff(theta(t),[t$2]) = -g*sin(theta(t)),
> theta(t), {theta(0)=0, D(theta)(0)=v[0]/l} );
```

$$\mathrm{DESol}(\{l\,(\frac{\partial^2}{\partial t^2}\theta(t))+g\sin(\theta(t))\},\{\theta(t)\},\{\mathrm{D}(\theta)(0)=\frac{v_0}{l},\theta(0)=0\})$$

```
> series(",t);
```

$$\frac{v_0}{l}\,t-\frac{1}{6}\frac{g\,v_0}{l^2}\,t^3+\frac{1}{120}\frac{g\,v_0\,(g+\frac{{v_0}^2}{l})}{l^3}\,t^5+\mathrm{O}(t^6)$$

Polinomok esetében eltűnik az ordó szimbólum, legalábbis ha a csonkítás rendje elég nagy:

```
> polynomial := a*x^3+b*x^2+c*x+d;
```

$$polynomial := a\,x^3 + b\,x^2 + c\,x + d$$

```
> taylor_series := series( polynomial, x, 4 );
```

$$taylor_series := d + c\,x + b\,x^2 + a\,x^3$$

```
> series( (x+y)^7, x, infinity ); # infinite order
```

$$y^7 + 7\,y^6\,x + 21\,y^5\,x^2 + 35\,y^4\,x^3 + 35\,y^3\,x^4 + 21\,y^2\,x^5 + 7\,y\,x^6 + x^7$$

Bár a sorfejtések polinomokra hasonlítanak (az utóbbi példákban különösen), a nekik megfelelő belső adatstruktúra, ahogy korábban láttuk, teljesen más.

```
> whattype( polynomial );
```

$$+$$

```
> op( polynomial );
```

$$a\,x^3,\, b\,x^2,\, c\,x,\, d$$

```
> whattype( taylor_series );
```

$$series$$

```
> op( taylor_series );
```

$$d,\, 0,\, c,\, 1,\, b,\, 2,\, a,\, 3$$

```
> op( 0, taylor_series ); # the main variable
```

$$x$$

A *series* típusú objektumokkal másként bánik a Maple; a polinomokra megengedett műveletek közül sok nem alkalmazható sorfejtésekre:

```
> sin_series := series( sin(x), x=0, 6 );
```

$$sin_series := x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 + \mathrm{O}(x^6)$$

```
> subs( x=2, sin_series );
```

```
Error, invalid substitution in series
```

```
> sin_series * sin_series;
```

$$(x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 + \mathrm{O}(x^6))^2$$

```
> expand(");
```

$$(x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 + \mathrm{O}(x^6))^2$$

```
> series( ", x );
```

$$x^2 - \frac{1}{3}\,x^4 + \mathrm{O}(x^6)$$

A fönti bonyodalmak egy részét elkerülhetjük a többváltozós Taylor sorfejtésekre szolgáló **mtaylor** eljárás alkalmazásával. Ennek oka az, hogy az **mtaylor** eredményének típusa nem *series*, hanem közönséges polinom:

```
> F := (3*x*y-2*y^2) * (3*x+y) / (6*x^2-6*y);
```

$$F := \frac{(3\,x\,y-2\,y^2)\,(3\,x+y)}{6\,x^2-6\,y}$$

```
> readlib( mtaylor ):
> mtaylor( F, [x=1,y], 5 );
```

$$\frac{3}{2}\,y+y^2-\frac{5}{2}\,(x-1)\,y^2+\frac{2}{3}\,y^3+4\,y^2\,(x-1)^2-\frac{23}{6}\,(x-1)\,y^3+\frac{2}{3}\,y^4$$

```
> normal(");
```

$$\frac{3}{2}\,y+\frac{15}{2}\,y^2-\frac{21}{2}\,x\,y^2+\frac{9}{2}\,y^3+4\,y^2\,x^2-\frac{23}{6}\,x\,y^3+\frac{2}{3}\,y^4$$

```
> subs( y=1, " );
```

$$\frac{85}{6}-\frac{43}{3}\,x+4\,x^2$$

Az egyváltozós függvények sorfejtésének reprezentációjára visszatérve a sorfejtések a **normal**-lal normalizálhatók, az egyes együtthatókat a **coeff**-fel nyerhetjük ki:

```
> series( F, x, 4 );
```

$$\frac{1}{3}\,y^2+\frac{1}{2}\,y\,x-\frac{1}{6}\,\frac{9\,y-2\,y^2}{y}\,x^2+\frac{1}{2}\,x^3+\mathrm{O}(x^4)$$

```
> normal(");
```

$$\frac{1}{3}\,y^2+\frac{1}{2}\,y\,x+(-\frac{3}{2}+\frac{1}{3}\,y)\,x^2+\frac{1}{2}\,x^3+\mathrm{O}(x^4)$$

```
> coeff( ", x^2 );
```

$$-\frac{3}{2}+\frac{1}{3}\,y$$

Csak a sorfejtés alapjául szolgáló „fő változó” szerinti együtthatókat kaphatjuk meg:

```
> coeff( "", y^2 );

Error, unable to compute coeff
```

A **coeftayl** eljárással megkaphatjuk a Taylor sorfejtés valamely együtthatóját anélkül, hogy magát a sorfejtést kiszámoltatnánk. Az

$$\mathbf{coeftayl}(f, x=x_0, k)$$

eljáráshívás ehelyett az f függvény $x=x_0$ körüli Taylor sorfejtésében az $(x-x_0)^k$ tag együtthatóját a

$$\lim_{x\to x_0}\frac{\frac{d^k f}{dx^k}(x)}{k!}$$

határértékként adja meg.

```
> readlib(coeftayl):
> coeftayl( F, x=0, 20 );
```

$$\frac{1}{3}\frac{1}{y^8}-\frac{3}{2}\frac{1}{y^9}$$

A **coeftayl** többváltozós sorokra is alkalmazható:

```
> coeftayl( F, [x,y]=[1,0], [1,3] );
```

$$\frac{-23}{6}$$

Ha az egyváltozós sorfejtésből csak a főtag érdekel bennünket, ezt explicit módon is kérhetjük:

```
> series( 1/tan(x), x );
```

$$x^{-1}-\frac{1}{3}x-\frac{1}{45}x^3+\mathrm{O}(x^4)$$

```
> series( leadterm( 1/tan(x) ), x, infinity );
```

$$x^{-1}$$

A sorfejtések differenciálhatók és integrálhatók:

```
> sin_series;
```

$$x-\frac{1}{6}x^3+\frac{1}{120}x^5+\mathrm{O}(x^6)$$

```
> diff( sin_series, x );
```

$$1-\frac{1}{2}x^2+\frac{1}{24}x^4+\mathrm{O}(x^5)$$

```
> integrate( sin_series, x );
```

$$\frac{1}{2}x^2-\frac{1}{24}x^4+\frac{1}{720}x^6+\mathrm{O}(x^7)$$

Sorok invertálása a **solve** eljárással végeztethető el:

```
> solve( y = sin_series, x );
```

$$y+\frac{1}{6}y^3+\frac{3}{40}y^5+\mathrm{O}(y^6)$$

Hasonlítsuk össze az alábbiakkal:

```
> arcsin_series := series( arcsin(y), y );
```

$$\mathit{arcsin_series} := y+\frac{1}{6}y^3+\frac{3}{40}y^5+\mathrm{O}(y^6)$$

és

```
> series( RootOf( y = sin(x), x ), y );
```

$$y+\frac{1}{6}y^3+\frac{3}{40}y^5+\mathrm{O}(y^7)$$

Az alkalmazott matematika egyik, sorok inverzének meghatározásával kapcsolatos problémája az

$$E = u + e\sin E$$

Kepler-egyenlet megoldásának sorfejtése. Itt E az „excentrikus anomáliát” jelöli, melyet az u főexcentricitás és az e pályaexcentricitás segítségével akarunk kifejezni (föltesszük, hogy e értéke kicsi, v. ö. [8]). A sorfejtés együtthatóit trigonometrikus függvények lineáris kombinációjaként írjuk föl:

```
> series( E - u - e*sin(E), E=u );
```

$$-e\sin(u) + (1 - e\cos(u))\,(E-u) + \frac{1}{2}\,e\sin(u)\,(E-u)^2 + \frac{1}{6}\,e\cos(u)$$
$$(E-u)^3 - \frac{1}{24}\,e\sin(u)\,(E-u)^4 - \frac{1}{120}\,e\cos(u)\,(E-u)^5 + \mathrm{O}((E-u)^6)$$

```
> solve( ", E - u );
```

$$\sin(u)\,e + \cos(u)\sin(u)\,e^2 + (\cos(u)^2\sin(u) - \frac{1}{2}\sin(u)^3)\,e^3 +$$
$$(\cos(u)^3\sin(u) - \frac{5}{3}\cos(u)\sin(u)^3)\,e^4 +$$
$$(\cos(u)^4\sin(u) - \frac{11}{3}\cos(u)^2\sin(u)^3 + \frac{13}{24}\sin(u)^5)\,e^5 + \mathrm{O}(e^6),$$
$$-\frac{e}{-1+e}\,u - \frac{1}{6}\,\frac{e}{(-1+e)^4}\,u^3 - \frac{1}{120}\,\frac{e\,(9\,e+1)}{(-1+e)^7}\,u^5 + \mathrm{O}(u^6)$$

Az első megoldásra van szükségünk, egyszerűsítsük is az E e szerinti sorfejtését:

```
> u + map( combine, "[1], 'trig' );
```

$$u + (\sin(u)\,e + \frac{1}{2}\sin(2\,u)\,e^2 + (\frac{3}{8}\sin(3\,u) - \frac{1}{8}\sin(u))\,e^3 +$$
$$(\frac{1}{3}\sin(4\,u) - \frac{1}{6}\sin(2\,u))\,e^4 +$$
$$(\frac{125}{384}\sin(5\,u) - \frac{27}{128}\sin(3\,u) + \frac{1}{192}\sin(u))\,e^5 + \mathrm{O}(e^6))$$

Számolás közben sokszor kényelmes lehet a sorfejtések polinommá konvertálása. Ez egyszerűen elvégezhető:

```
> sin_series;
```

$$x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 + \mathrm{O}(x^6)$$

```
> convert( sin_series, polynom );
```

$$x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5$$

A Padé-féle approximáció és a lánctörtek szerinti sorfejtés, vagyis racionális függvényekkel való közelítések ugyancsak rendelkezésünkre állnak a Maple rendszerben. Ehhez fölhasználhatjuk a **convert** eljárást vagy a numapprox csomag **pade** és **confracform** eljárásait (hacsak nem akarjuk először meghatározni magát a sorfejtést, vagy eredményként formula helyett nem eljárásra van szükségünk).

```
> convert( sin_series, ratpoly );
```

$$\frac{-\frac{7}{60}x^3 + x}{1 + \frac{1}{20}x^2}$$

```
> convert( sin_series, ratpoly, 1, 2);
```

$$\frac{x}{1 + \frac{1}{6}x^2}$$

```
> convert( sin_series, 'confrac' );
```

$$\frac{x}{1 + \frac{x^2}{6 - \frac{7}{10}x^2}}$$

Ezek a közelítések használhatók az úgynevezett Csebisev sorfejtésekkel együtt is:

```
> Digits := 5:
> chebyshev( sin(x), x ); # Chebyshev expansion
```

$$.88010\,\mathrm{T}(1, x) - .039127\,\mathrm{T}(3, x) + .00049952\,\mathrm{T}(5, x) - .30152\,10^{-5}\,\mathrm{T}(7, x)$$

```
> convert( ", ratpoly );
```

$$\frac{.89083\,\mathrm{T}(1, x) - .027887\,\mathrm{T}(3, x)}{\mathrm{T}(0, x) + .025529\,\mathrm{T}(2, x)}$$

```
> with( orthopoly, T ): "";
```

$$\frac{.97449\,x - .11155\,x^3}{.97447 + .051058\,x^2}$$

A Maple a polinom-konverziókat igen liberálisan kezeli:

```
> series( sin(x+a)/x^2, x, 4 );
```

$$\sin(a)\,x^{-2} + \cos(a)\,x^{-1} - \frac{1}{2}\sin(a) - \frac{1}{6}\cos(a)\,x + \mathrm{O}(x^2)$$

```
> convert( ", polynom );
```

$$\frac{\sin(a)}{x^2} + \frac{\cos(a)}{x} - \frac{1}{2}\sin(a) - \frac{1}{6}\cos(a)\,x$$

```
> series( sqrt(x*(1-x)) , x=0 , 2 );
```

$$\sqrt{x} - \frac{1}{2}x^{3/2} - \frac{1}{8}x^{5/2} + \mathrm{O}(x^{7/2})$$

```
> convert( ", polynom );
```

$$\sqrt{x} - \frac{1}{2}x^{3/2} - \frac{1}{8}x^{5/2}$$

```
> (a*x^3+b*x^2)/(x^2+1);
```

$$\frac{a\,x^3 + b\,x^2}{x^2+1}$$

```
> series( ", x=infinity, 8 ); # asymptotic expansion
```

$$a\,x + b - \frac{a}{x} - \frac{b}{x^2} + \frac{a}{x^3} + \frac{b}{x^4} - \frac{a}{x^5} - \frac{b}{x^6} + \frac{a}{x^7} + \mathrm{O}(\frac{1}{x^8})$$

```
> convert( ", polynom );
```

$$a\,x + b - \frac{a}{x} - \frac{b}{x^2} + \frac{a}{x^3} + \frac{b}{x^4} - \frac{a}{x^5} - \frac{b}{x^6} + \frac{a}{x^7}$$

A Stirling formula az általános típusú *aszimptotikus sorfejtések* egyik példája:

```
> asympt( ln(x!), x, 4 );
```

$$(\ln(x) - 1)\,x + \frac{1}{2}\ln(x) + \ln(\sqrt{2}\,\sqrt{\pi}) + \frac{1}{12}\frac{1}{x} + \mathrm{O}(\frac{1}{x^3})$$

A **series** parancs helyett az **asympt** eljárást alkalmaztuk a végtelenben vett sorfejtésre:

```
> simplify( ", ln ); # simplification of logarithmic terms
```

$$(\ln(x) - 1)\,x + \frac{1}{2}\ln(x) + \frac{1}{2}\ln(2) + \frac{1}{2}\ln(\pi) + \frac{1}{12}\frac{1}{x} + \mathrm{O}(\frac{1}{x^3})$$

[3] szerint a számítógépes algebrai rendszerek alkalmazásának egyik nehézsége az, hogy a rendszer nem tudja a megfelelő helyen alkalmazni az összes rendelkezésére álló matematikai tudásanyagot. Ezt illusztrálja a következő példa:

$$\lim_{\epsilon\to 0} \frac{\epsilon}{\sqrt{1+\epsilon}\,\sin^2\phi + \sqrt{1-\epsilon}\,\cos^2\phi - 1}$$

A Maple és sok más rendszer nem képes a $\sin^2\phi + \cos^2\phi = 1$ trigonometrikus egyszerűsítési szabályt a kellő pillanatban alkalmazni, s emiatt hibás eredményre jut:

```
> expr := epsilon / ( sqrt(1+epsilon)*sin(phi)^2 +
>   sqrt(1-epsilon)*cos(phi)^2-1 );
```

$$expr := \frac{\varepsilon}{\sqrt{1+\varepsilon}\sin(\phi)^2 + \sqrt{1-\varepsilon}\cos(\phi)^2 - 1}$$

```
> limit( expr, epsilon=0 );
```

$$0$$

A válasz rossz. A probléma gyökere a következő hibás sorfejtés:

```
> series( expr, epsilon, 3 );
```

$$\frac{1}{\sin(\phi)^2 + \cos(\phi)^2 - 1}\varepsilon - \frac{\frac{1}{2}\sin(\phi)^2 - \frac{1}{2}\cos(\phi)^2}{(\sin(\phi)^2 + \cos(\phi)^2 - 1)^2}\varepsilon^2 + \mathrm{O}(\varepsilon^3)$$

Az első tag nevezője nullára egyszerűsíthető.

Szerencsére a Maple **series** parancsa megenged egy kis „finomhangolást" a 0 együtthatók fölismerésénél. Csupán a **Testzero** környezeti változót kell átdefiniálnunk, ami a rendszerben a viszonylag triviális

```
proc() evalb(normal(args[1]) = 0 ) end
```

eljárással van inicializálva. Az ajánlott változtatás: a **normal** helyett használjuk a **Normalizer** környezeti változót, és úgy definiáljuk át ezt az eljárást, hogy a **series** jobban kezelje az együtthatókat:

```
> Testzero := proc() evalb( Normalizer(args[1]) = 0 ) end:
> Normalizer := proc() normal( simplify(args[1]) ) end:
```

A Maple-t így adaptálva már minden nagyszerűen működik:

```
> readlib( forget ):
> forget( series ): # clear the remember table of series
> forget( limit ): # clear the remember table of limit
> series( expr, epsilon, 3 );
```

$$\frac{1}{\frac{1}{2}\sin(\phi)^2 - \frac{1}{2}\cos(\phi)^2} + 2\,\frac{-\frac{1}{8}\sin(\phi)^2 - \frac{1}{8}\cos(\phi)^2}{(2\cos(\phi)^2 - 1)\,(\frac{1}{2}\sin(\phi)^2 - \frac{1}{2}\cos(\phi)^2)}\,\varepsilon + \mathrm{O}(\varepsilon^2)$$

```
> map( combine, ", 'trig' );
```

$$-\frac{2}{\cos(2\,\phi)} + \frac{1}{\cos(4\,\phi) + 1}\,\varepsilon + \mathrm{O}(\varepsilon^2)$$

```
> limit( expr, epsilon=0 );
```

$$\frac{2}{\sin(\phi)^2 - \cos(\phi)^2}$$

11.2. Függvényközelítések

A legtöbb matematikai függvény kiértékeléséhez először valamely könnyen kiszámítható közelítését kell előállítanunk. Általában mind a hely, mind az időigény szempontjából táblázatok tárolásánál és az ezekkel végzett interpolációnál hatékonyabbak az analitikus közelítések. Különböző alakú közelítő függvények léteznek: Taylor polinomok, Padé közelítések, Csebisev sorok stb. (v. ö. [155]). A `numaprox` csomag numerikus közelítések kifejlesztésére szolgáló különféle eljárásokat tartalmaz. Ebben a részben az exponenciális függvényre próbálunk jó közelítéseket találni a $(-2, 2)$ intervallumon, ennek kapcsán meg fogjuk vizsgálni ezeket a módszereket. További példákat [77]-ben találhat az érdeklődő Olvasó.

Első közelítésünk a 0 körüli ötödrendű Taylor polinom:

```
> with( numapprox ): # load the numapprox package
> Digits := 4: # set precision for approximation
```

```
> taylor( exp(x), x );
```

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\mathrm{O}(x^6)$$

```
> P[5] := convert( ", polynom );
```

$$P_5 := 1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5$$

A polinom Horner alakra konvertálható.

```
> hornerform( P[5], x );
```

$$1+(1+(\frac{1}{2}+(\frac{1}{6}+(\frac{1}{24}+\frac{1}{120}\,x)\,x)\,x)\,x)\,x$$

```
> taylorapprox := unapply(",x);
```

$$\mathit{taylorapprox} := x \to 1+(1+(\frac{1}{2}+(\frac{1}{6}+(\frac{1}{24}+\frac{1}{120}\,x)\,x)\,x)\,x)\,x$$

A Taylor-féle maradéktagból adódó hibabecslés

$$e^x - P_5(x) = \frac{1}{720}e^{\xi}x^6$$

ahol ξ valamely 0 és x közti érték. A hiba részletesebb vizsgálatára fölhasználhatjuk a **maximize** és a **minimize** eljárásokat:

```
> readlib(maximize)( exp(x) - P[5], x, -2..2 );
```

$$e^{(-2)} - \frac{1}{15}$$

```
> evalf(");
```

$$.06863$$

```
> readlib(minimize)( exp(x) - P[5], x, -2..2 );
```

$$e^{\%1} - 1 - \%1 - \frac{1}{2}\%1^2 - \frac{1}{6}\%1^3 - \frac{1}{24}\%1^4 - \frac{1}{120}\%1^5$$
$$\%1 :=$$
$$\mathrm{RootOf}(24\,e^{_Z} - 24 - 24\,_Z - 12\,_Z^2 - 4\,_Z^3 - _Z^4,\ -.01787)$$

```
> evalf(");
```

$$.00001125$$

Másik lehetőségünk a numaprox csomag **infnorm** eljárásának alkalmazása a hiba kiszámítására. Az eljárás az $[a, b]$ intervallumon folytonos f függvény

$$\|f\|_\infty = \max_{a\le x\le b}|f(x)|$$

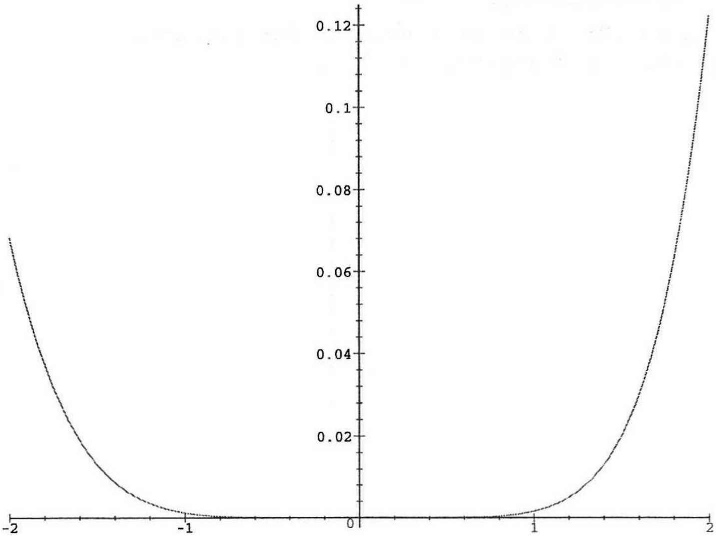
végtelen normájának közelítését számolja ki.

```
> infnorm( exp - taylorapprox, -2..2 );
```

$$.1224$$

A maximális hiba tehát közelítőleg 0.122. A 11.2. ábráról látható, hogy a hiba eloszlása nem egyenletes a $(-2, 2)$ intervallumon.

```
> Digits := 10: # reset precision for plotting
> plot( exp - taylorapprox, -2..2 );
```



11.2. ábra: Az exponenciális függvény 0 körüli ötödrendű Taylor polinomjának hibagörbéje

Ezután a (3,2) rendű Padé közelítést vizsgáljuk:

```
> Digits := 4: # set precision for approximation
> R[3,2] := pade( exp(x), x, [3,2] );
```

$$R_{3,2} := \frac{1 + \frac{3}{5}x + \frac{3}{20}x^2 + \frac{1}{60}x^3}{1 - \frac{2}{5}x + \frac{1}{20}x^2}$$

A szorzások száma minimalizálható, ha lánctört alakra konvertálunk:

```
> confracform(",x);
```

$$\frac{1}{3}x + \frac{17}{3} + \frac{152}{3}\,\frac{1}{x - \frac{117}{19} + \frac{3125}{361}\,\frac{1}{x - \frac{35}{19}}}$$

```
> padeapprox := unapply(",x);
```

$$padeapprox := x \to \frac{1}{3}x + \frac{17}{3} + \frac{152}{3}\,\frac{1}{x - \frac{117}{19} + \frac{3125}{361}\,\frac{1}{x - \frac{35}{19}}}$$

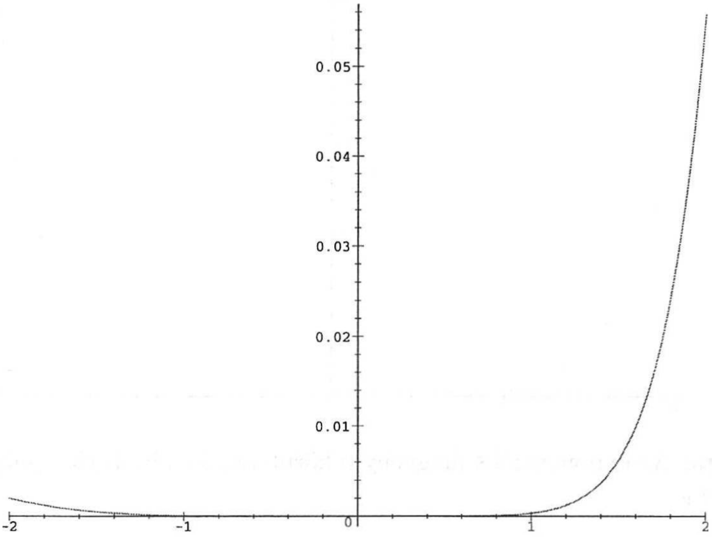
A hiba kisebb a Taylor közelítésénél:

```
> infnorm( exp - padeapprox, -2..2 );
```

$$.05572$$

A maximális hiba tehát közelítőleg 0.056. A hiba eloszlását a $(-2, 2)$ intervallumon a 11.3. ábra mutatja.

```
> Digits := 10: # reset precision for plotting
> plot( exp - padeapprox, -2..2 );
```



11.3. ábra: Az exponenciális függvény 0 körüli (3, 2) rendű Padé közelítésének hibagörbéje

Ha jobb közelítéseket szeretnénk nyerni a $(-2, 2)$-n, az x^n hatványok szerinti kifejtések helyett használhatunk ortogonális polinomokat. A numaprox csomagban Csebisev polinomokkal dolgozhatunk a **chebyshev** és a **chebpade** eljárások segítségével. További segédfüggvények a **chebmult**, **chebsort** és a **chebdeg**.

A 11.4. és a 11.5. ábra az exponenciális függvény Csebisev, illetve Csebisev–Padé közelítésének hibagörbéjét mutatja.

```
> Digits := 4: # set precision for approximation
> chebP[5] := chebyshev( exp(x), x );
```

$$\mathit{chebP}_5 := 1.266\,\mathrm{T}(0, x) + 1.131\,\mathrm{T}(1, x) + .2715\,\mathrm{T}(2, x) + .04434\,\mathrm{T}(3, x) + .005475\,\mathrm{T}(4, x) + .0005430\,\mathrm{T}(5, x)$$

```
> chebP[5] := (eval@subs)( T = orthopoly[T], '' );
```

$$\mathit{chebP}_5 := 1.000 + 1.001\,x + .4992\,x^2 + .1665\,x^3 + .04380\,x^4 + .008688\,x^5$$

```
> hornerform(",x);
```

$$1.000 + (1.001 + (.4992 + (.1665 + (.04380 + .008688\,x)\,x)\,x)\,x)\,x$$

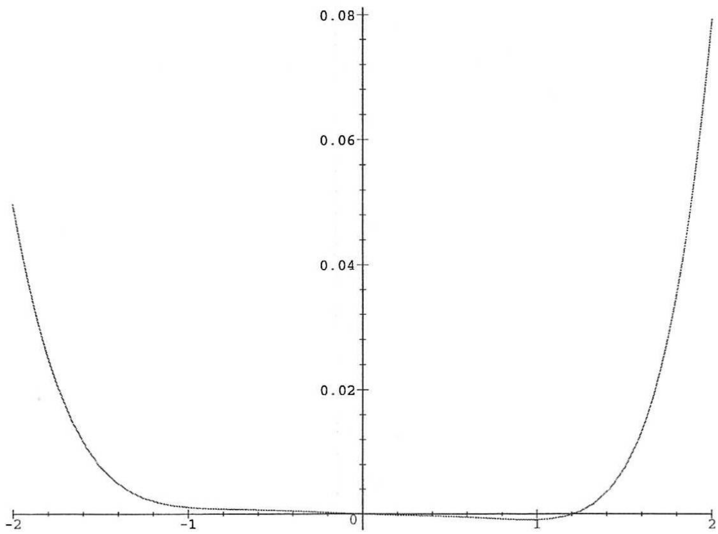
```
> chebapprox := unapply(",x);
```

$$chebapprox := x \to 1.000 + (1.001 + (.4992 + (.1665 + (.04380 + .008688\,x)\,x)\,x)\,x)\,x$$

```
> infnorm( exp - chebapprox, -2..2 );
```

$$.07944$$

```
> Digits := 10:  # reset precision for plotting
> plot( exp - chebapprox, -2..2 );
```



11.4. ábra: Az exponenciális függvény 0 körüli ötödfokú Csebisev közelítésének hibagörbéje

```
> Digits := 4:  # set precision for approximation
> chebR[3,2] := chebpade( exp(x), x=-2..2, [3,2] );
```

$$chebR_{3,2} := \frac{1.208\,\mathrm{T}(0, \frac{1}{2}x) + 1.223\,\mathrm{T}(1, \frac{1}{2}x) + .2894\,\mathrm{T}(2, \frac{1}{2}x) + .03149\,\mathrm{T}(3, \frac{1}{2}x)}{\mathrm{T}(0, \frac{1}{2}x) - .7092\,\mathrm{T}(1, \frac{1}{2}x) + .08160\,\mathrm{T}(2, \frac{1}{2}x)}$$

```
> chebR[3,2] := eval( subs( T = orthopoly[T], " ) );
```

$$chebR_{3,2} := \frac{.9186 + .5643\,x + .1447\,x^2 + .01575\,x^3}{.9184 - .3546\,x + .04080\,x^2}$$

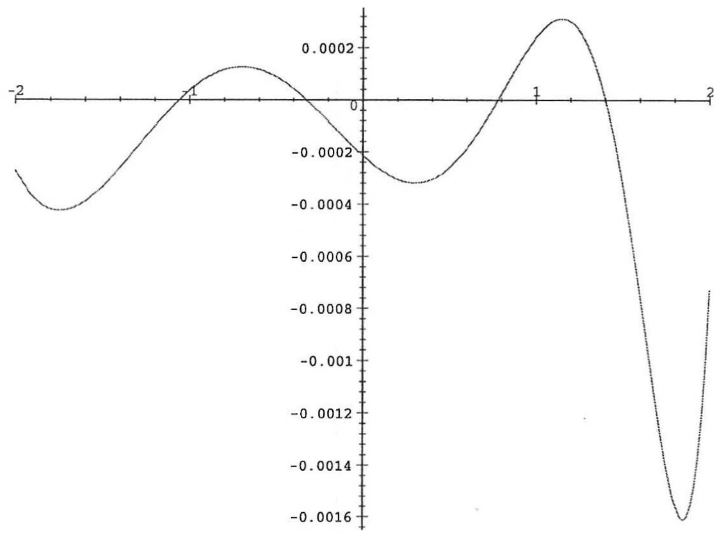
```
> Digits := 5:  confracform("",x);
```

$$.38603\,x + 6.9017 + \cfrac{65.125}{x - 6.6514 + \cfrac{8.9424}{x - 2.0398}}$$

```
> chebpadeapprox := unapply(",x):
> infnorm( exp - chebpadeapprox, -2..2 );
```

$$.0016120$$

```
> Digits := 10:  # reset precision for plotting
> plot( exp - chebpadeapprox, -2..2 );
```



11.5. ábra: Az exponenciális függvény 0 körüli ötödfokú Csebisev–Padé közelítésének hibagörbéje

Az exponenciális függvény fönti legjobb egyenletes közelítésének hibagörbéje váltakozó előjelű. A hibafüggvény „kisimításával” tovább javíthatunk rajta. Ezt a **minimax** eljárással tehetjük meg, amely a Remez algoritmus (l. [161,162]) segítségével keresi meg az adott súlyfüggvényhez és fokszámhoz tartozó legjobb minimax racionális közelítést. Részletesebben elmondva a **minimax** adott $[a,b]$ intervallumon értelmezett w pozitív súlyfüggvényhez és f folytonos valós függvényhez kiszámolja azt a p/q racionális függvényt, amelyre

$$\max_{a \leq x \leq b} w(x)|f(x) - p(x)/q(x)|$$

minimális az összes olyan p/q racionális függvény halmazában, amelyek p számlálója legföljebb m-ed fokú, q nevezője pedig legföljebb n-ed fokú polinom.

Először tekintsük az exponenciális függvény ötödfokú polinommal való minimax közelítését:

```
> minimaxP[5] := minimax( exp, -2..2, [5,0], 1,
>    'maxerror' );
```

$$minimaxP_5 := x \to 1.003137731 + (1.002686425 + (.4860498437 + (.1624711341 + (.05072464720 + .01005370263\, x)\, x)\, x)x)x$$

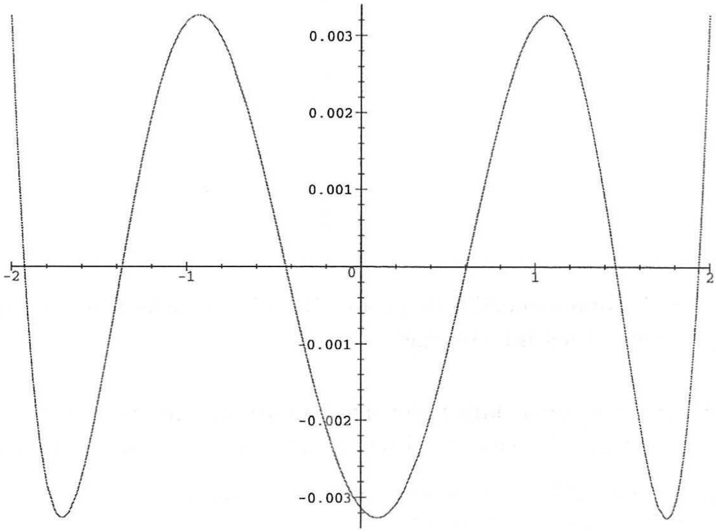
Az ötödfokú polinommal való minimax közelítés maximális hibáját is megadta a Remez algoritmus:

```
> maxerror;
```

$$.0032642362$$

A hibagörbe a 11.6. ábrán látható.

```
> plot( exp - minimaxP[5], -2..2 );
```



11.6. ábra: Az exponenciális függvény ötödfokú polinommal való minimax közelítésének hibagörbéje

Ezután az exponenciális függvény $(3,2)$ fokú racionális függvénnyel való legjobb minimax közelítését határozzuk meg. A hibagörbét a 11.7. ábrán láthatjuk.

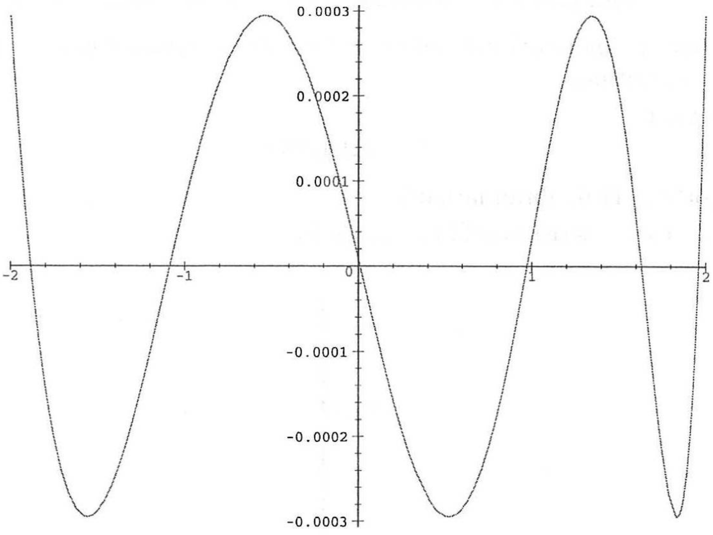
```
> minimaxR[3,2] := minimax( exp, -2..2, [3,2], 1,
>    'maxerror' );
```

$$minimaxR_{3,2} := x \to (.9223619741 + (.5751247262 + (.1517495935 + .01744849612\, x)\, x)\, x) / (.9223699852 + (-.3480389375 + .03881500741\, x)\, x)$$

```
> maxerror;
```

$$.000294578$$

```
> plot( exp - minimaxR[3,2], -2..2 );
```



11.7. ábra: Az exponenciális függvény $(3,2)$ fokú racionális függvénnyel való minimax közelítésének hibagörbéje

Ezt a részt az exponenciális függvénynek a relatív hibára nézve legjobb olyan minimax közelítésével zárjuk, ahol $(3,2)$ fokú racionális függvénnyel közelítünk:

```
> minimaxapprox32 := minimax( exp, -2..2, [3,2],
>    x -> 1/ exp(x), 'maxerror' );
```

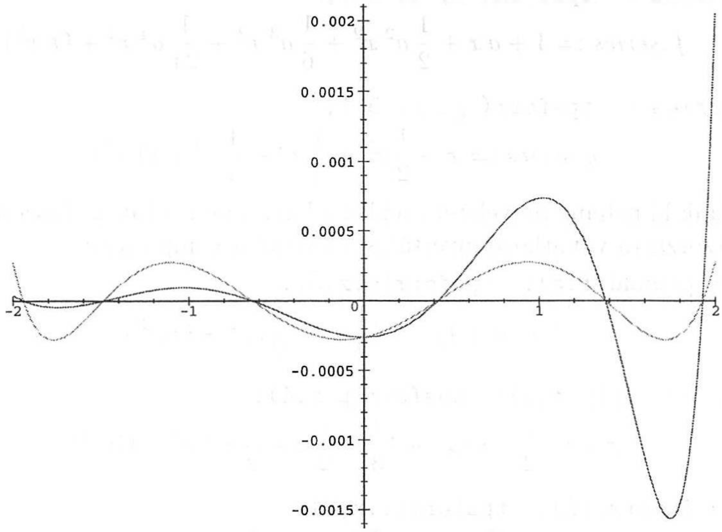
$$\mathit{minimaxapprox32} := x \to (.9133842488 + (.5504989644 + (.1363414316 + .01407497787\,x)\,x)\,x)\,/(.9131465849 + (-.3625452263 + .04342670756\,x)\,x)$$

```
> maxerror;
```

$$.0002764958030$$

A 11.8. ábrán együtt láthatjuk a mostani és az előző, az abszolút hibára nézve vett minimax közelítés hibagörbéjét.

```
> plot( { 1 - minimaxapprox32 / exp,
>    exp - minimaxapprox32 }, -2..2 );
```



11.8. ábra: Az exponenciális függvény (3, 2) fokú racionális függvényekkel való minimax közelítéseinek hibagörbéje

11.3. Hatványsorok

A `powseries` csomagban találunk formális hatványsorok manipulációjára használható eljárásokat. Mint az összes többi Maple csomagot, ezt is be kell töltenünk:

```
> with( powseries );
```

$$[compose, evalpow, inverse, multconst, multiply, negative, powadd, powcos, powcreate, powdiff, powexp, powint, powlog, powpoly, powsin, powsolve, powsqrt, quotient, reversion, subtract, tpsform]$$

A legtöbbet példák révén tudunk meg a csomagról. Mostantól fogva az

$$\exp(ax) = \sum_{n=0}^{\infty} \frac{a^n}{n!} x^n \quad \text{és az} \quad \ln(1+x) = \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{n} x^n$$

hatványsorokat tekintjük. Először is definiáljuk a **powcreate** eljárással a két hatványsort:

```
> powcreate( f(n)=a^n/n! ):
> powcreate( g(n)=(-1)^(n+1)/n, g(0)=0 ):
```

Most azokat a szabályokat definiáltuk, amelyek segítségével a hatványsorok együtthatói meghatározhatók. Kérdezzük meg az első öt tagot a **tpsform** (**t**runcated **p**ower **s**eries **form**) eljárással:

```
> f_series := tpsform( f, x, 5 );
```

$$f_series := 1 + a\,x + \frac{1}{2}\,a^2\,x^2 + \frac{1}{6}\,a^3\,x^3 + \frac{1}{24}\,a^4\,x^4 + O(x^5)$$

```
> g_series := tpsform( g, x, 5 );
```

$$g_series := x - \frac{1}{2}\,x^2 + \frac{1}{3}\,x^3 - \frac{1}{4}\,x^4 + O(x^5)$$

Próbáljunk ki néhány műveletet ezekkel a hatványsorokkal: az összeadást, szorzást, a szorzásra vonatkozó invertálást és végül a kompozíciót:

```
> s := powadd(f,g):  tpsform(s,x,3);
```

$$1 + (a+1)\,x + (\frac{1}{2}\,a^2 - \frac{1}{2})\,x^2 + O(x^3)$$

```
> p := multiply(f,g):  tpsform(p,x,4);
```

$$x + (-\frac{1}{2} + a)\,x^2 + (\frac{1}{3} - \frac{1}{2}\,a + \frac{1}{2}\,a^2)\,x^3 + O(x^4)$$

```
> i := inverse(f):  tpsform(i,x,5);
```

$$1 - a\,x + \frac{1}{2}\,a^2\,x^2 - \frac{1}{6}\,a^3\,x^3 + \frac{1}{24}\,a^4\,x^4 + O(x^5)$$

```
> p := multiply(i,f):  tpsform(p,x,10);
```

$$1 + O(x^{10})$$

```
> c := compose(f,g):  tpsform(c,x,4);
```

$$1 + a\,x + (-\frac{1}{2}\,a + \frac{1}{2}\,a^2)\,x^2 + (\frac{1}{3}\,a - \frac{1}{2}\,a^2 + \frac{1}{6}\,a^3)\,x^3 + O(x^4)$$

```
> a := 1:  # special case, which may take a while
> c := compose(f,g):  eval( tpsform(c,x,10) );
```

$$1 + x + O(x^{10})$$

Az utolsó eredményt hatékonyabban is megkaphattuk volna:

```
> c := powexp(g):  tpsform(c,x,10);
```

$$1 + x + O(x^{10})$$

A hatványsorokra alkalmazható a logaritmus függvény:

```
> a := 'a':  # reset a to a free variable
> r := powlog(f):  tpsform(r,x,10);
```

$$a\,x + O(x^{10})$$

További műveletek a differenciálás és az integrálás:

```
> d := powdiff(f):  tpsform(d,x,4);
```

$$a + a^2\,x + \frac{1}{2}\,a^3\,x^2 + \frac{1}{6}\,a^4\,x^3 + O(x^4)$$

```
> i := powint(f):  tpsform(i,x,4);
```

$$x + \frac{1}{2}\,a\,x^2 + \frac{1}{6}\,a^2\,x^3 + O(x^4)$$

A hatványsorok kompozícióra vonatkozó inverzének meghatározása is könnyen elvégezhető a Maple rendszerben. Az $\ln(1+x)$ függvény hatványsorát használjuk példaként; inverz hatványsorként az $\exp(x)-1$ függvény hatványsorát kell kapnunk:

```
> r := reversion(g): tpsform(r,x,5);
```

$$x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\mathrm{O}(x^5)$$

```
> c := compose(g,r): tpsform(c,x,5);
```

$$x+\mathrm{O}(x^5)$$

```
> c := compose(r,g): tpsform(c,x,5);
```

$$x+\mathrm{O}(x^5)$$

Befejezésül egy közismert példa: számítsuk ki az égi mechanikában az elliptikus mozgások bizonyos sorfejtéseinél használt f és g sorokat. (V. ö. [63].) A sorfejtés együtthatóit a következő rekurziók definiálják:

$$f_n=-\mu g_{n-1}-\sigma(\mu+2\epsilon)\frac{\partial f_{n-1}}{\partial\epsilon}+(\epsilon-2\sigma^2)\frac{\partial f_{n-1}}{\partial\sigma}-3\mu\sigma\frac{\partial f_{n-1}}{\partial\mu},\ f_0=1,$$

$$g_n=f_{n-1}-\sigma(\mu+2\epsilon)\frac{\partial g_{n-1}}{\partial\epsilon}+(\epsilon-2\sigma^2)\frac{\partial g_{n-1}}{\partial\sigma}-3\mu\sigma\frac{\partial g_{n-1}}{\partial\mu},\qquad g_0=0.$$

Először a koordináta-függvényeket definiáljuk:

```
> mu := (mu,sigma,epsilon) -> mu:
> sigma := (mu,sigma,epsilon) -> sigma:
> epsilon := (mu,sigma,epsilon) -> epsilon:
```

A **D** operátor segítségével megadhatók a rekurrens egyenletek:

```
> with( powseries ):
> powcreate( f(n) = -mu*g(n-1) - sigma*(mu+2*epsilon)
>    *D[3](f(n-1)) + (epsilon-2*sigma^2)*D[2](f(n-1))
>    - 3*mu*sigma*D[1](f(n-1)), f(0) = 1 ):
> powcreate( g(n) = f(n-1) - sigma*(mu+2*epsilon)
>    *D[3](g(n-1)) + (epsilon-2*sigma^2)*D[2](g(n-1))
>    - 3*mu*sigma*D[1](g(n-1)), g(0) = 0 ):
```

és kiszámíthatók a sorfejtések első tagjai:

```
> tpsform(f,T,5);
```

$$1-\mu T^2+3\,\mu\,\sigma\,T^3+(\mu^2+3\,(\varepsilon-2\,\sigma^2)\,\mu-9\,\mu\,\sigma^2)\,T^4+\mathrm{O}(T^5)$$

Mindennek azonban van egy hátránya. Az Olvasó valószínűleg a részeredményeket is kifejtett formában szeretné látni, de mivel a rekurrens relációkat a **powcreate** paranccsal definiáltuk, ez nehezen érhető el. A sor kiszámítása után egyszerűsíthetjük a végeredményt:

```
> map(factor,");
```

$$1-\mu T^2+3\,\mu\,\sigma\,T^3+\mu\,(\mu+3\,\varepsilon-15\,\sigma^2)\,T^4+\mathrm{O}(T^5)$$

A részeredmények kifejtése a `powseries` csomag implementációjának részleteit is kihasználó következő trükkel lehetséges: mielőtt bármilyen csonkított sorfejtést végeztetnénk a **tpsform**-mal, adjuk ki a következő utasításokat:

```
> f(_k) := 'expand'( f(_k) ): g(_k) := 'expand'( g(_k) ):
```

Ezután valóban kifejtett formában jelennek meg a részeredmények.

11.4. Határértékek

A Maple gyakran használ általánosított sorfejtést határértékek meghatározásához. (Lásd [72, 80].) Például az

```
> ln(x) - ln(x+exp(-x));
```

$$\ln(x) - \ln(x + e^{(-x)})$$

```
> series( ", x=infinity, 2 );
```

$$-\frac{1}{x\,e^x} + \frac{O(\frac{1}{x^2})}{(e^x)^2}$$

asszimptotikus sorfejtés alapján világos, hogy

```
> Limit( "", x=infinity ): " = value(");
```

$$\lim_{x\to\infty} \ln(x) - \ln(x + e^{(-x)}) = 0$$

Itt a **value** eljárással értékeljük ki a **Limit** tétlen eljárás hívásának eredményét. Világosan látszik, hogy nem kell sokat törődnünk a részletekkel, csak alkalmazni kell a **limit** eljárást.

A **limit**-nek megadhatunk néhány opciót: `left`, `right`, `real` és `complex`. Egy példa:

```
> Limit( cos(x)^(1/x^3), x=0 ): " = value(");
```

$$\lim_{x\to 0} \cos(x)^{(\frac{1}{x^3})} = \mathit{undefined}$$

```
> Limit( cos(x)^(1/x^3), x=0, 'right' ): " = value(");
```

$$\lim_{x\to 0+} \cos(x)^{(\frac{1}{x^3})} = 0$$

```
> Limit( cos(x)^(1/x^3), x=0 , 'left' ): " = value(");
```

$$\lim_{x\to 0-} \cos(x)^{(\frac{1}{x^3})} = \infty$$

Ha a **limit**-nek nem mondjuk meg, hogy melyik oldali határértékre vagyunk kiváncsiak, a (valós) kétoldali határértéket határozza meg (kivéve a $+\infty$-ben vagy a $-\infty$-ben vett határértékeket, ahol természetesen a bal, illetve a jobb oldali határérték értendő).

Néhány esetben a Maple-nek több információra van szüksége. Például:

```
> y := exp(-a*x)*cos(b*x);
```

$$y := e^{(-a\,x)} \cos(b\,x)$$

```
> limit( y, x = infinity );
```

$$\lim_{x\to\infty} e^{(-a\,x)} \cos(b\,x)$$

Természetesen a Maple nem tételezte föl, hogy mi az a változót pozitívnak képzeltük. De megadhattuk volna az

```
> assume( a>0 ):
```

föltételt a határérték kiszámítása előtt.

```
> limit( y, x = infinity );
```

$$0$$

Ahogy a fejezet elején megjegyeztük, a legtöbb határérték meghatározása sorok segítségével történik. Emiatt néha szükség lehet az `Order` környezeti változó értékének megnövelésére:

```
> expr := ( exp(x) - sum( x^k/k!, k=0..11 ) ) / x^12;
```

$$expr := (e^x - 1 - x - \frac{1}{2}x^2 - \frac{1}{6}x^3 - \frac{1}{24}x^4 - \frac{1}{120}x^5 - \frac{1}{720}x^6 - \frac{1}{5040}x^7 - \frac{1}{40320}x^8 - \frac{1}{362880}x^9 - \frac{1}{3628800}x^{10} - \frac{1}{39916800}x^{11})/x^{12}$$

```
> limit( expr, x=0 );
```

$$\lim_{x\to 0}(e^x - 1 - x - \frac{1}{2}x^2 - \frac{1}{6}x^3 - \frac{1}{24}x^4 - \frac{1}{120}x^5 - \frac{1}{720}x^6 - \frac{1}{5040}x^7 - \frac{1}{40320}x^8 - \frac{1}{362880}x^9 - \frac{1}{3628800}x^{10} - \frac{1}{39916800}x^{11})/x^{12}$$

```
> Order := 13:
> limit( expr, x=0 );
```

$$\frac{1}{479001600}$$

A csonkított sorfejtésekről szóló első alfejezetben már láttunk példát olyan sorfejtésre, ahol a helyes eredmények eléréséhez módosítanunk kellett a `Testzero` zérustesztelő függvényt. Most egy másik ugyanilyen jellegű példa következik:

```
> expr := x / ( sin(phi)^2 +(1+x)*cos(phi)^2 - 1);
```

$$expr := \frac{x}{\sin(\phi)^2 + (1+x)\cos(\phi)^2 - 1}$$

```
> limit( ", x=0 );
```

$$0$$

A kifejezés valójában független x-től, de értéke biztosan nem 0:

```
> simplify( expr );
```

$$\frac{1}{\cos(\phi)^2}$$

A rossz eredmény oka a 0 körüli hibás csonkított sorfejtés:

```
> series( expr, x=0, 3 );
```

$$\frac{1}{\sin(\phi)^2+\cos(\phi)^2-1}\,x-\frac{\cos(\phi)^2}{(\sin(\phi)^2+\cos(\phi)^2-1)^2}\,x^2+\mathrm{O}(x^3)$$

Az első tag nevezője 0-ra egyszerűsíthető, de ezt a Maple automatikusan nem veszi észre. A Maple a sorfejtések főtagjának meghatározására két eljárást használ, a **Testzero**-t és a **Normalizer**-t. A **Testzero** környezeti változó, amely a viszonylag triviális

```
proc() evalb(normal(args[1]) = 0 ) end
```

eljárással van inicializálva, azt határozza meg, hogy a főegyüttható az osztásra nézve nulla-e. A **Normalizer** környezeti változót a **series** az osztók főtagjainak normalizálására használja. A helyes eredmények eléréséhez csak ezt a két eljárást kell átdefiniálnunk. Az ajánlott módosítás: a **Testzero**-ban a **normal** helyett a **Normalizer** környezeti változót használjuk, és definiáljuk át úgy ezt az eljárást, hogy a **series** jobban kezelje az együtthatókat.

```
> Testzero := proc() evalb( Normalizer(args[1]) = 0 ) end:
> Normalizer := proc() normal( simplify(args[1]) ) end:
```

A Maple fönti módosítása után már minden jól működik.

```
> readlib( forget ):
> forget( series ):  # clear the remember table of series
> forget( limit ):  # clear the remember table of limit
> series( expr, x=0, 3 ), limit( expr, x=0 );
```

$$\frac{1}{\cos(\phi)^2},\ \frac{1}{\cos(\phi)^2}$$

11.5. Gyakorlatok

1. Vizsgáljuk a szinusz függvény különféle numerikus közelítéseit a $[-\pi, \pi]$ intervallumon.

2. Vizsgáljuk az

$$x \longmapsto \frac{1}{x}\int_0^x \frac{\sin(\sin u)}{u}\,du$$

függvény különféle numerikus közelítéseit, ha $|x| < \pi$.

3. Számoljuk ki a következő határértékeket, és ellenőrizzük az eredményeket.

a) $\displaystyle\lim_{x\to 0}\frac{\sin x}{x}$

b) $\displaystyle\lim_{x\to 0}(\sin x)^{1/x}$

c) $\displaystyle\lim_{x\to 0}\frac{1-\cos x}{x}$

d) $\lim_{x\to\infty} \left(1+\frac{\pi}{x}\right)^x$

e) $\lim_{x\to 0} x^{\sin x}$

f) $\lim_{x\to\infty} (2^x + 3^x)^{1/x}$

4. Számoljuk ki a következő határértékeket:

a) $\lim_{x\to\infty} \frac{\ln x}{x}$

b) $\lim_{x\to\infty} \frac{\ln x}{e^x}$

c) $\lim_{x\to\infty} \frac{x^2+\sin x}{2x^2+\cos 4x}$

d) $\lim_{x\downarrow 0} \frac{2}{1+e^{-1/x}}$

e) $\lim_{x\to\infty} \sinh(\tanh x) - \tanh(\sin x)$

5. Egy új Maple szekcióban vagy a **restart** parancs kiadása után számítsuk ki az $x^3 - 1 - 4x^2 + 5x$ polinom $x = 1$ pont körüli tizedrendű Taylor sorfejtését. Ügyeljünk arra, hogy a polinom tagjait ebben a sorrendben gépeljük be. Milyen típusú lesz a nyert kifejezés? Mit fog gondolni a Maple arról, hogy meddig megy a sorfejtés?

6. Mi a $\binom{2n}{n}$ asszimptotikus sorfejtése?

7. Mi a Lambert-féle W függvény hatványsorfejtése?

8. Ebben a gyakorlatban a $T_n(x)$ elsőfajú Csebisev polinomokat vezetjük be generálófüggvényük segítségével.

(a) Számítsuk ki az $\frac{1-t^2}{1-2xt+t^2}$ Taylor sorfejtését $x = 0$ és $t = 0$ körül nyolcadrendig.

(b) A $T_n(x)$ elsőfajú Csebisev polinomokat az

$$\frac{1-t^2}{1-2xt+t^2} = \sum_{n=0}^{\infty} \epsilon_n T_n(x) t^n$$

generátorfüggvény definiálja, ahol $\epsilon_0 = 1$ és $\epsilon_n = 2$, ha $n \geq 1$. Számítsuk ki a $T_2(x)$ és a $T_{10}(x)$ polinomot. Ellenőrizzük az **orthopoly[T]** beépített paranccsal számításainkat.

9. Határozzuk meg az $E = u+e\sin E$ Kepler egyenlet megoldásának huszonötöd rendű Taylor sorfejtését a **RootOf** eljárás segítségével. Vessük össze ennek az eljárásnak a hatékonyságát a jelen fejezet első alfejezetében leírt eljárásával.

10. Keressük meg az

$$x \longmapsto \frac{x^a - a^x}{x^x - a^a}$$

függvény $x = a$ körüli harmadrendű sorfejtését.

11. A Josephson-féle áramkör vizsgálata során szükségünk lehet a következő módon definiált ptan függvény sorfejtésére: $\mathrm{ptan}(s) = p$, ha $p - \tan p = s$. Számítsuk ki ennek a függvénynek a sorfejtését.

12. Hasonlítsuk össze az $E = u + e\sin E$ Kepler egyenletnek a fejezet első alfejezetében sorokkal előállított megoldását a Bessel függvényekkel fölírt egzakt megoldással: $E = u + 2\sum_{n=1}^{\infty} J_n(n\,e)\,\sin(n\,u)/n$.

12. Összetett adattípusok

A Maple adattípusainak aktív ismeretére gyakran még akkor is szükségünk van, ha a Maple-t csak interaktívan, szimbolikus kalkulátorként használjuk. Emlékezzünk vissza a *polynom* és a *randpoly* elemi adattípusokra és az ilyen kifejezések kezelésére, illetve egyszerűsítésére. Ez a fejezet megismerteti az Olvasót az olyan *összetett adattípusokkal*, mint például a *sequence*, *set*, *list* és az *array*, amelyek elemi adattípusokból épülnek föl, és objektumok csoportosítására használhatók.

12.1. Sorozat

Már korábban is találkoztunk `exprseq` (expression sequence)[1] típusú objektumokkal. Például az `op` függvény eredménye a legtöbb esetben objektumok vesszővel elválasztott sorozata:

```
> polynomial := x^3 - 6*x^2 + 11*x - 6;
```

$$polynomial := x^3 - 6\,x^2 + 11\,x - 6$$

```
> sequence := op( polynomial );
```

$$sequence := x^3,\ -6\,x^2,\ 11\,x,\ -6$$

Sorozatokat gyakran használ a Maple. Nemcsak a `solve(polynomial,x)` parancs eredménye sorozat, hanem maguk a függvényhívás argumentumai is sorozatot alkotnak az alábbi példában:

```
> arguments := polynomial, x;
```

[1] A továbbiakban a „kifejezéssorozat" és a „sorozat" elnevezéseket azonos értelemben használjuk, tehát nem (csak) számsorozatokról lesz szó. (*A Fordító megjegyzése.*)

$$arguments := x^3 - 6\,x^2 + 11\,x - 6,\ x$$

```
> whattype( arguments );
```

$$exprseq$$

```
> solve( arguments );
```

$$1,\ 2,\ 3$$

Eljárásdefiníció belsejében az aktuális argumentumok sorozatára az args változóval hivatkozhatunk; az aktuális argumentumok számát az nargs változó tartalmazza. Ezt illusztrálja a következő példa.

```
> procargs := proc()
>   print( `number of arguments` = nargs,
>       `actual arguments` = args )
> end:
> procargs(x);
```

$$number\ of\ arguments = 1,\ actual\ arguments = (x)$$

```
> procargs(x,y,z);
```

$$number\ of\ arguments = 3,\ actual\ arguments = (x,\ y,\ z)$$

Figyeljük meg, hogy a sorozat egy objektum; belső reprezentációja egyetlen adatvektor (12.1. ábra).

exprseq	↑expr1	↑expr2	↑expr3	↑expr4	......

12.1. ábra: A sorozatnak megfelelő belső adatsruktúra

Az ábra ↑expr1, ↑expr2, ... szimbólumai az expr1, expr2 stb. kifejezéseknek megfelelő adatvektorokra mutató pointereket jelentenek. A kifejezéssorozat egyes komponenesei nem föltétlenül azonos típusú kifejezések.

```
> seq1 := M, a, p, l, e:
> seq2 := 77, 97, 112, 108, 101:  # name in ASCII code
> `concatenated sequence` := seq1, seq2;
```

$$concatenated\ sequence := M,\ a,\ p,\ l,\ e,\ 77,\ 97,\ 112,\ 108,\ 101$$

Az üres sorozat jelölésére a Maple a NULL speciális nevet használja:

```
> `empty sequence`:= NULL;
```

$$empty\ sequence :=$$

```
> 1, 2, `empty sequence`, 2, 1;
```

$$1,\ 2,\ 2,\ 1$$

Sorozatokat előállíthatunk a **seq** függvénnyel is:

```
> # generate the first nine odd prime numbers
> seq( ithprime(i), i = 2..10 );
```

$$3,\ 5,\ 7,\ 11,\ 13,\ 17,\ 19,\ 23,\ 29$$

Az **seq**($f(i), i = m...n$) alakú függvényhívás által generált sorozat $f(m)$, $f(m+1)$, $f(m+2), \ldots, f(n)$. Ennek alternatívájaként a $ sorozat-operátor is rendelkezésünkre áll a Maple-ben.

```
> x$4;
```

$$x, x, x, x$$

A **diff** függvény hívásakor a fenti módon való használatát kivéve azonban a sorozat-operátor alkalmazási lehetőségei eléggé korlátozottak, sőt még veszélyes is lehet.

```
> ('[k,l] $ k=1..2') $ l=3..4;
```

$$[1, 3], [2, 3], [1, 4], [2, 4]$$

Ebben a példában az idézőjelek föltétlenül szükségesek az idő előtti kiértékelés megelőzésére, és még több idézőjelre lenne szükség, ha a k és l sorozatindexekhez esetleg már korábban értéket rendeltünk volna. A legrosszabb esetben ilyen rémes parancsot kellene begépelnünk:

```
> ('['k','l'] $ 'k'=1..2') $ 'l'=3..4:
```

Az Olvasó ehelyett bizonyára a következő formát részesítené előnyben:

```
> seq( seq( [i,j], i = 1..2 ), j = 3..4 ):
```

Az **seq**($f(i), i =$ *expression*) alakú függvényhívás által generált sorozatot a Maple úgy állítja elő, hogy az *expression* kifejezés minden operandusára alkalmazza f-et:

```
> seq( i^2, i = { 1, 2, 3, 4, 5 } );
```

$$1, 4, 9, 16, 25$$

```
> seq( i^2, i = x + y + z );
```

$$x^2, y^2, z^2$$

A sorozat egy elemének kiválasztását a [ ] szelekciós operátorral végezhetjük el:

```
> "[2]; # 2nd element
```

$$y^2$$

```
> ""[-1]; # last element
```

$$z^2$$

Természetesen egyidejűleg a sorozat több elemét is kiválaszthatjuk:

```
> sequence := v, w, x, y, z:  sequence[ 2..4 ];
```

$$w, x, y$$

Az utolsó utasítást úgy tekinthetjük, mint a következő utasítás rövidítését:

```
> seq( sequence[i], i = 2..4 );
```

$$w, x, y$$

Ilyen rövidítések máshol is előfordulnak, például amikor konkatenációval akarjuk generálni nevek *p1, p2,* ... sorozatát.

```
> p.( 1..5 );
```

$$p1, p2, p3, p4, p5$$

```
> seq( p.i, i = 1..5 );
```

$$p1, p2, p3, p4, p5$$

Ezt a paragrafust a

$$left_expression .. right_expression$$

alakú részkifejezésekre vonatkozó rövid megjegyzéssel zárjuk. Ez egy szabályos `range` típusú Maple kifejezés. Láttunk már ilyen kifejezéseket, amikor az integrálás és az összegzés határait adtuk meg. Akármilyen Maple kifejezést használhatunk a *left_expression* és a *right_expression* megadására, de a legtöbb esetben ezeknek numerikus értékre kell kiértékelődniük. Ennek kapcsán megjegyezzük, hogy a *range* operátort kettőnél több ponttal is jelölhetjük. De emlékeztetünk arra, hogy egy pont a konkatenációs operátor jele.

12.2. Halmaz

Leggyakrabban akkor használunk `set` (halmaz) típusú objektumokat, amikor a Maple **solve** eljárásával egyenletrendszereket próbálunk megoldani. Ennek az eljárásnak az argumentumai: egyenletek egy halmaza és ismeretlenek egy halmaza; a megoldás pedig általában `set` típusú objektumok sorozata.

A Maple halmazokra az általánosan használt matematikai jelölést alkalmazza: kapcsos zárójelbe zárt sorozatot. A halmazban egy adat egynél többször nem fordul elő. Felhasználóként szinte semmi befolyásunk sincs a halmaz elemeinek sorrendjére, mivel a rendszer a belső címek szerinti rendezést használ.

```
> { 1, 3, 5, 2, 4}; # used earlier in increasing ordering
```

$$\{1, 2, 3, 4, 5\}$$

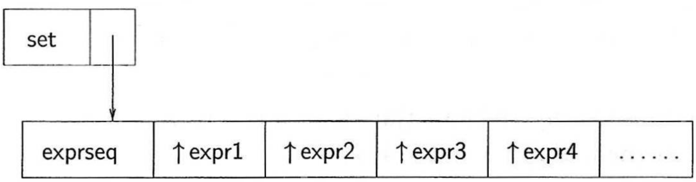
```
> { x, x, x*y, x*(x-1), y*x, x^2-x };
```

$$\{x, x\,y, x\,(x-1), x^2-x\}$$

```
> `empty set` := {};
```

$$empty\ set := \{\}$$

A 12.2. ábrán látható a halmazok belső reprezentációja.



12.2. ábra: A halmaznak megfelelő belső adatsruktúra

A kifejezéssorozat tagjai tárolási címük szerint növekvő sorrendbe rendezettek. Ez biztosítja, hogy minden elem csak egyszer forduljon elő.

A *set* típusú objektumokra a szokásos műveletek: **union**, **minus** és **intersect** alkalmazhatók.

```
> {0,1,2,3} union {0,2,4,6};
```

$$\{0, 1, 2, 3, 4, 6\}$$

```
> {0,1,2,3} minus {0,2,4,6};
```

$$\{1, 3\}$$

```
> {0,1,2,3} intersect {0,2,4,6};
```

$$\{0, 2\}$$

Megkérdezhetjük, hogy egy bizonyos Maple objektum eleme-e egy halmaznak, és ha igen, akkor mi a halmazbeli pozíciója.

```
> member( 2, {0,1,2,3}, 'position' );
```

$$true$$

```
> position;
```

$$3$$

A halmaz egy elemét a [ ] szelekciós operátorral vagy az **op** függvénnyel választhatjuk ki. Az utóbbi kiválasztási séma kevésbé hatékony, mivel az egész halmazt kiértékeli:

```
> # generate all subsets of 1,2,3
> collection := combinat[powerset](3);
```

$$collection := \{\{\}, \{1, 3\}, \{1, 2, 3\}, \{2, 3\}, \{3\}, \{1\}, \{2\}, \{1, 2\}\}$$

```
> nops( collection ); # number of elements in the set
```

$$8$$

```
> collection[4]; # 4th element in set
```

$$\{2, 3\}$$

```
> collection[-1]; # last element in set
```

$$\{1, 2\}$$

```
> op( 8, collection );
```

$$\{1, 2\}$$

```
> collection[6..8]; # returns a subset
```

$$\{\{1\}, \{2\}, \{1, 2\}\}$$

```
> collection[3..-3]; # omit first and last 2 elements
```

$$\{\{1, 2, 3\}, \{2, 3\}, \{3\}, \{1\}\}$$

```
> collection[]; # sequence of set elements
```

$$\{\}, \{1, 3\}, \{1, 2, 3\}, \{2, 3\}, \{3\}, \{1\}, \{2\}, \{1, 2\}$$

Ha a halmaz bizonyos föltételeknek megfelelő elemeit egyszerre akarjuk kiválasztani, sokat segíthet a a Maple **select** eljárása. Létezik az ezzel ellentétes hatású **remove** függvény is. Nézzünk néhány példát.

```
> die := rand(-10..10): numberset := { 'die()' $ 11 };
```

$$numberset := \{0, -4, -5, -6, 6, 7, -8, 8, 10\}$$

```
> select( isprime, numberset ); # select prime numbers
```

$$\{7\}$$

```
> # select nonnegative integers
> select( type, numberset, 'nonnegint' );
```

$$\{0, 6, 7, 8, 10\}$$

```
> remove( x -> x < -3, numberset ); # remove numbers < -3
```

$$\{0, 6, 7, 8, 10\}$$

A kiválasztás és az eltávolítás általános alakja:

select(*criterion, set, extra arguments*)

és

remove(*criterion, set, extra arguments*).

A megadott kiválasztási föltételnek a *true* vagy a *false* logikai értéket kell eredményezni. A kiválasztási föltétel további extra argumentumait mindig a halmaz után kell paraméterként megadni.

12.3. Lista

Míg a halmazok kapcsos zárójelek közé írt sorozatok, lista típusú objektumot úgy konstruálhatunk, hogy a sorozatot szögletes zárójelbe zárjuk. Már láttunk ilyen objektumokat, amikor a **collect** eljárásban a polinom változóinak sorrendjét meghatározó argumentumokat adtunk meg:

```
> 1 + x + y^2 + z^3 + x^2*y^2 + x^2*z^3 + y^3*z^3;
```

$$1 + x + y^2 + z^3 + x^2\,y^2 + x^2\,z^3 + y^3\,z^3$$

```
> collect( ", [ x, y, z ] );
```

$$(y^2 + z^3)\,x^2 + x + 1 + y^2 + z^3 + y^3\,z^3$$

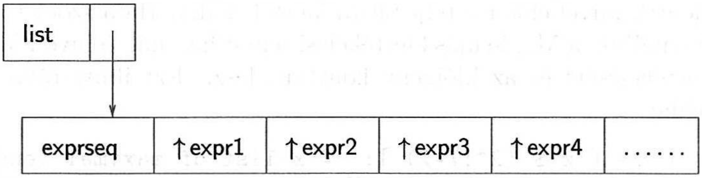
```
> collect( ", [ z, y, x ] );
```

$$(x^2 + 1 + y^3)\,z^3 + (x^2 + 1)\,y^2 + x + 1$$

A *list* és a *set* típusú objektumok közti fontos különbségek az alábbbiak:

- a listában ugyanaz az objektum többször is előfordulhat;
- a lista megőrzi az elemek sorrendjét;
- a lista elemeinek értéke könnyen megváltoztatható.

A lista objektumok belső fölépítése (v. ö. 12.3. ábra) azonban hasonlít a halmazokéra (lásd a 12.2. ábrát).



12.3. ábra: A listának megfelelő belső adatsruktúra

```
> [ x, x, x*y, x*(x-1), y*x, x^2-x ];
```

$$[x,\, x,\, x\, y,\, x\,(x-1),\, x\, y,\, x^2 - x]$$

```
> sort( ", 'address' ); # sort by machine address
```

$$[x,\, x,\, x\, y,\, x\, y,\, x\,(x-1),\, x^2 - x]$$

```
> convert( ", set );
```

$$\{x,\, x\, y,\, x\,(x-1),\, x^2 - x\}$$

```
> [ x$5 ]; # list of five x'ses
```

$$[x,\, x,\, x,\, x,\, x]$$

```
> `empty list` := [];
```

$$\mathit{empty\ list} := []$$

Az alábbiakban fölsoroljuk a leggyakrabban használt listaműveleteket, és példákat mutatunk arra, hogyan hajthatók végre ezek a Maple rendszerben.

- Lista hosszának meghatározása

```
> cl := [ black, red, green, yellow, blue, white ]:
> nops( cl ); # number of colors
```

$$6$$

- Elem keresése listában

```
> member( indigo, cl );
```

$$\mathit{false}$$

```
> member( blue, cl, 'position' );
```

$$\mathit{true}$$

```
> position;
```

$$5$$

- Egy vagy több elem kiválasztása

```
> cl[5]; # efficient selection
```

$$\mathit{blue}$$

```
> op(5,cl); # inefficient selection
```

$$\mathit{blue}$$

A listaelem **op**-pal való kiválasztásának időigénye $O(n)$ (n a lista hosszát jelenti), mivel ekkor a teljes lista kiértékelődik. Ha a szögletes zárójeleket használjuk, a Maple más kiértékelési sémát használ, (elnyomja a teljes lista kiértékelését) és az időigény konstans lesz. Ezt illusztrálja a következő példa:

```
> L := [ x $ (2^17-2) ]: # x-list of maximal length
> # access-time to "long" list
> time(): L[1]: time(): cpu_time = ("-""") * second;
```

$$cpu_time = 0$$

```
> time(): op(1,L): time(): cpu_time = ("-""") * second;
```

$$cpu_time = .081\,second$$

```
> time(): L[2^17-2]: time():
> cpu_time = ("-""") * second;
```

$$cpu_time = 0$$

```
> time(): op(2^17-2,L): time():
> cpu_time = ("-""") * second;
```

$$cpu_time = .079\,second$$

A listaelemek kiválasztása még általánosabban is elvégezhető. Néhány, az előző cl listával kapcsolatos példa:

```
> cl[-1]; # first list element from the right
```

white

```
> cl[3..6]; # return a sublist
```

[*green*, *yellow*, *blue*, *white*]

```
> op(3..6,cl); # return a sequence
```

green, *yellow*, *blue*, *white*

```
> cl[-2..-1]; # the sublist of last two elements
```

[*blue*, *white*]

```
> cl[]; # sequence of list elements
```

black, *red*, *green*, *yellow*, *blue*, *white*

```
> cl[ 2 .. nops(cl) ];
```

[*red*, *green*, *yellow*, *blue*, *white*]

```
> cl[ 2 .. -1 ];
```

[*red*, *green*, *yellow*, *blue*, *white*]

```
> subsop( 1 = NULL, cl );
```

[*red*, *green*, *yellow*, *blue*, *white*]

```
> cl[ 1 .. nops(cl)-1 ];
```

[*black*, *red*, *green*, *yellow*, *blue*]

```
> cl[ 1 .. -2 ];
```

[*black*, *red*, *green*, *yellow*, *blue*]

```
> subsop( nops(cl) = NULL, cl );
```

[*black*, *red*, *green*, *yellow*, *blue*]

- Elem hozzáfűzése a lista elejéhez, végéhez és elem beszúrása a listába

```
> [ crimson, op(cl) ];
```

$$[crimson, black, red, green, yellow, blue, white]$$

```
> [ op(cl), crimson ];
```

$$[black, red, green, yellow, blue, white, crimson]$$

```
> [ op(cl[1..3]), crimson, op(cl[4..nops(cl)]) ];
```

$$[black, red, green, crimson, yellow, blue, white]$$

Megjegyezzük, hogy hosszú listák esetében az **op** direkt használatánál hatékonyabb egy részlista létrehozása és sorozattá konvertálása, mivel ekkor nem kell az egész listát kiértékelni.

- Listák konkatenációja

```
> cl2 := [ crimson, indigo ]:  [ op(cl), op(cl2) ];
```

$$[black, red, green, yellow, blue, white, crimson, indigo]$$

- Listaelem helyettesítése

```
> subsop( 5 = indigo, cl ); # cl unchanged
```

$$[black, red, green, yellow, indigo, white]$$

```
> cl[5] := purple; cl; # cl changed
```

$$cl_5 := purple$$

$$[black, red, green, yellow, purple, white]$$

Az utóbbi értékadás hatékonyabb, mint a `cl:=subsop(5=indigo,cl)`. De a listaelemeknek közvetlenül csak legföljebb 100 hosszúságú listákban adhatunk értéket.

- Az összes megadott listaelem egyidejű helyettesítése

```
> subs( red = crimson, cl ); # replace every color red
```

$$[black, crimson, green, yellow, purple, white]$$

- Listák átrendezése

```
> # sort the list (in lexicographic ordering)
> sort( cl, lexorder ); # variable cl is not changed
```

$$[black, green, purple, red, white, yellow]$$

```
> # reverse the list
> [ seq( cl[-i], i=1..nops(cl) ) ];
```

$$[white, purple, yellow, green, red, black]$$

```
> # rotate the list one position to the left or right
> [ op( 2..nops(cl), cl ), cl[1] ];
```

$$[red, green, yellow, purple, white, black]$$

```
> # rotate the list one position to the left or right
> [ op( cl[2..-1] ), cl[1] ];
```

$$[red, green, yellow, purple, white, black]$$

```
> [ cl[nops(cl)], op( 1..nops(cl)-1, cl ) ];
```
$$[white, black, red, green, yellow, purple]$$
```
> [ cl[-1], op( cl[1..-2] ) ];
```
$$[white, black, red, green, yellow, purple]$$

Néhány alapvető Maple függvény és konstrukció fölhasználásával olyan könnyedén végrehajthatjuk a legtöbb listaműveletet, hogy a Maple rendszer tervezői nem vesztegették az idejüket arra, hogy külön függvényneveket vezessenek be ezekre a műveletekre. Ha óhajtjuk, könnyedén átírhatjuk a fenti példákat Maple eljárásokká. Például az utolsó két művelet így írható föl eljárásként:

```
> rotateleft := proc(L::list)
>   [ op( L[2..-1] ), L[1] ]
> end:
> rotateright := proc(L::list)
>   [ L[-1], op( L[1..-2] ) ]
> end:
> rotateleft( cl );
```
$$[red, green, yellow, purple, white, black]$$
```
> (rotateright@@3)( cl ); # rotate 3 places to the right
```
$$[yellow, purple, white, black, red, green]$$
```
> (rotateleft@rotateright)( cl ); # do nothing
```
$$[black, red, green, yellow, purple, white]$$

Az Olvasó megkérdezhetné, vajon miért az `[op(L),elem]` paranccsal fűztük hozzá a lista végéhez az `elem` új listaelemet, miért nem használtuk inkább az `L[nops(L)+1]:=elem` értékadást. Az alábbi példából kiderül, hogy csak létező listaelemeknek adhatunk értéket; a lista nem bővíthető ezen a módon:

```
> L := [a,b,c];
```
$$L := [a, b, c]$$
```
> L[3] := [c,c]; # assignment of existing list element
```
$$L_3 := [c, c]$$
```
> L;
```
$$[a, b, [c, c]]$$
```
> L[3,2] := d; # assignment of existing list element
```
$$L_{3,2} := d$$
```
> L;
```
$$[a, b, [c, d]]$$
```
> L[4] := e; # no extension of list

Error, out of bound assignment to a list
```

A 12.1. táblázatban összegezzük a T összetett adatsruktúrára alkalmazható szelektorokat, ahol T típusa *sequence, set* vagy *list* lehet.

Kiválasztási mód	**A visszaadott adatstruktúra**
`T[]`	a komponensekből álló kifejezéssorozat
`T[i]`, (`i`$\in \mathbb{N}$)	balról az i-dik komponens
`T[-i]`, (`i`$\in \mathbb{N}$)	jobbról az i-dik komponens
`T[i..j]`, (`i,j`$\in \mathbb{Z}$)	az i és j közti tartományba eső komponensek sorozatként, halmazként vagy listaként
`T[i,j,...]`	`T[i][j,...]`
`T[i..j,k,...]`	`seq( (T[n])[k,...], k=i..j )` mint sorozat, halmaz vagy lista

12.1. táblázat: Kiválasztás sorozatból, halmazból és listából

12.4. Tömb

Ez az alfejezet rövid bevezetés az *array* adattípus és a vele kapcsolatos objektumok, például a *vektorok* és a *mátrixok* használatába. A 18. fejezetben részletesebben ismertetjük a tömbök alkalmazásait a mátrixalgebrában. Ebben a részben csupán ezen adatstruktúrák megkonstruálására szorítkozunk. A tömbök kiértékeléséről szó lesz még a 12.6. alfejezetben.

A konvencionális programozási nyelvekhez hasonlóan a Maple-ben is léteznek az egy-, két- és többdimenziós tömbök, mint adatstruktúrák. Az alábbi példában tömb típusú objektumot hozunk létre:

```
> array( -1..1, 0..1,
>    [ (-1,0)=a, (-1,1)=b, (0,0)=c, (0,1)=d ] );
```

$$\begin{array}{l}\mathrm{array}(-1..1, 0..1, [\\ \quad(-1, 0) = a\\ \quad(-1, 1) = b\\ \quad(0, 0) = c\\ \quad(0, 1) = d\\ \quad(1, 0) = ?_{1,0}\\ \quad(1, 1) = ?_{1,1}\\ \quad])\end{array}$$

Először a kétdimenziós tömb indexeinek tartományát specifikáltuk. Másodszor néhány, de nem föltétlenül az összes tömbelemhez értéket rendeltünk. Legyünk óvatosak, hogy ne keverjük össze a kerek és a szögletes zárójeleket. Ez egy kicsit zavaró, mivel a tömbelemek kiválasztása szögletes zárójellel történik.

Az *array* adattípus valójában a következő alfejezetben tárgyalt *table* adattípus speciális esete, melyet maga a Maple is gyakran használ (például eljárások emlé-

kezőtábláiban). Az *array* adattípus jellemző vonása, hogy indexek halmazaként egész számok folytonos szegmenseinek Descartes-féle szorzatát engedi meg.

Első látásra az *array* adattípus hasonló a *list* adattípushoz. Ez abban is tükröződik, ahogy *list* típusú objektumokból *array* típusú objektumokat konstruálhatunk. Az alábbiakban egy-, két- és háromdimenziós tömböket hozunk létre.

```
> v := array( [1+a,2+b,3+c] );
```

$$v := [1 + a, 2 + b, 3 + c]$$

```
> type( v, list );
```

false

```
> type( v, vector );
```

true

```
> type( v, array );
```

true

```
> M := array( [ [1-p,2-q], [1-r,2-s] ] );
```

$$M := \begin{bmatrix} 1-p & 2-q \\ 1-r & 2-s \end{bmatrix}$$

```
> A := array( [ [ [1,2], [3,4] ], [ [5,6], [8,9] ],
>    [ [10,11], [12,13] ] ] );
```

$A :=$ array(1..3, 1..2, 1..2, [
(1, 1, 1) = 1
(1, 1, 2) = 2
(1, 2, 1) = 3
(1, 2, 2) = 4
(2, 1, 1) = 5
(2, 1, 2) = 6
(2, 2, 1) = 8
(2, 2, 2) = 9
(3, 1, 1) = 10
(3, 1, 2) = 11
(3, 2, 1) = 12
(3, 2, 2) = 13
])

A tömbelemek listaként (vagy listák listájaként) való beírásával egy csomó gépelést spórolunk meg. Továbbá ha nem adjuk meg az indexhatárokat, a Maple a lista hosszából ezeket is megállapítja (azt föltételezve, hogy az indexhatárok 1-gyel kezdődnek). Ez jól látható a háromdimenziós tömböt definiáló példánk outputjának elején is. Ilyen egytől kezdődő indexezésű egy- és kétdimenziós tömböket legtöbbször vektorokként és mátrixokként használunk, a Maple ennyiben is a tipikus vektor- és mátrixjelöléseket alkalmazza.

```
> type( M, matrix );
```

$$true$$

Az előző `M` mátrixot és `v` vektort – a tömbök reprezentációjának megfelelő formában – létrehozhattuk volna a **matrix** és a **vektor** parancsokkal is:

```
> M := matrix( [ [ 1-p, 2-q ], [ 1-r, 2-s ] ] );
```

$$M := \begin{bmatrix} 1-p & 2-q \\ 1-r & 2-s \end{bmatrix}$$

```
> v := vector( [ 1+a, 2+b, 3+c ] );
```

$$v := [1+a, 2+b, 3+c]$$

Néha csak deklarálni akarunk egy vektort anélkül, hogy megadnánk elemeit. A Maple számára ez sem probléma:

```
> vec := vector(100);
```

$$vec := \mathrm{array}(1..100, [\,])$$

```
> vec := array(-1000..1000, [] );
```

$$vec := \mathrm{array}(-1000..1000, [\,])$$

Mátrixok konstruálásának további kényelmes módja az indexfüggvények megadása. Egy 4×4-es Hilbert mátrix például így hozható létre:

```
> h := (i,j) -> 1/(i+j-x); # index function
```

$$h := (i, j) \to \frac{1}{i+j-x}$$

```
> matrix( 4, 4, h );
```

$$\begin{bmatrix} \frac{1}{2-x} & \frac{1}{3-x} & \frac{1}{4-x} & \frac{1}{5-x} \\ \frac{1}{3-x} & \frac{1}{4-x} & \frac{1}{5-x} & \frac{1}{6-x} \\ \frac{1}{4-x} & \frac{1}{5-x} & \frac{1}{6-x} & \frac{1}{7-x} \\ \frac{1}{5-x} & \frac{1}{6-x} & \frac{1}{7-x} & \frac{1}{8-x} \end{bmatrix}$$

Használhattuk volna a `linalg` csomag beépített **linalg[hilbert]** függvényét is. Másik példánk egy zérusmátrix megkonstruálása:

```
> matrix( 3, 3, 0 );
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ebben a függvényhívásban a 0 az azonosan 0 függvényt jelenti.

Az **array** eljárásnak vannak a speciális alakú mátrixokkal kapcsolatos opciói is:

```
> M := array( symmetric, 1..2, 1..2 ): M[1,2]:=1:
> print(M);
```

$$\begin{bmatrix} M_{1,1} & 1 \\ 1 & M_{2,2} \end{bmatrix}$$

```
> v := array( sparse, 1..10 ): v[1]:=1: v[5]:=-1:
> print(v);
```

$$[1, 0, 0, 0, -1, 0, 0, 0, 0, 0]$$

```
> M := array( sparse, symmetric, 1..2, 1..2 ): M[1,2]:=1:
> print(M);
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

További opciók az `antisymmetric`, `identity` és a `diagonal`. Ezek az opciók csak az **array** (és mint később látni fogjuk, a **table**) eljárásban használhatók, a **vector** vagy **matrix** eljárásokban nem. Az opciók szerepe annak jelzése, hogyan számíthatók ki az egyes mátrixelemek anélkül, hogy mindegyiket tárolni kellene a tömbnek megfelelő belső adatstruktúrában. Ha egynél több opciót adtunk meg, a rendszer balról jobbra haladva egyenként használja föl őket.

A Maple-ben bevezethetünk „új” indexfüggvényeket is. Példaként a `tridiag` speciális indexfüggvényt definiáljuk, ez valójában az **index/tridiag** eljárás megadásának felel meg. Ha az **index/tridiag** eljárást két argumentummal hívjuk, kiértékel egy tömbelemet; három argumentumot megadva értékadást végez a megfelelő elemnek (lásd a 12.2. táblázatot).

Hívás formátuma	**Célja**
`‘index/tridiagonal ‘( ind, arr )`	kiértékelés
`‘index/tridiagonal ‘( ind, arr, val )`	értékadás

12.2. táblázat: A speciális indexfüggvény különböző hívásai

Itt `ind` az indexeket tartalmazó lista, `arr` az indexezett tömb, `val` pedig az értékadásban fölhasználandó objektumokat tartalmazó lista. Az `arr` első argu-

mentuma nevek sorozata, amelyekből az első, most `tridiagonal`, azt mutatja meg, hogy melyik indexfüggvényt kell használni. Ezt a rendszer „lenyeli", ezért a tábla első argumentuma a maradék nevekből áll, itt többek között `symmetric` vagy `antisymmetric` állhat.

Ha az **index/tridiag**-ot két argumentummal hívjuk, a kiválasztott tömbelemet adja vissza. Ha három argumentumot adunk meg, azt várjuk el a függvénytől, hogy adjon értéket a megfelelő tömbelemnek. A visszaadott érték az értékadás eredménye lesz. Az ilyenfajta hívás tipikus mellékhatásaként megváltozik az `arr` értéke is:

```
> `index/tridiagonal` := proc( ind, arr, val )
>   if nargs = 2 then
>     if outofband( ind )
>     then 0
>     else arr[op(ind)]
>     fi
>   elif nargs = 3 then
>     if outofband( ind )
>       then ERROR(`band width 1 is exceeded`,
>            'indices' = ind )
>       else arr[op(ind)] := op(val)
>       fi
>   else ERROR(`invalid arguments`)
> quad fi
> end:
```

A következő rutin azt teszteli, hogy az index a megadott 1 szélességű határok között van-e:

```
> outofband := proc( ind::list )
>   local indset, pairs, p, d:
>   indset := {op(ind)};
>   if nops(indset) <= 1 then RETURN(false) fi;
>   pairs := combinat[choose]( indset, 2 );
>   for p in pairs do
>     d := p[1] - p[2];
>     if type( d, integer ) and abs(d) > 1
>     then RETURN(true)
>     fi
>   od:
>   RETURN(false)
> end:
```

Néhány példa:

```
> A := array( tridiagonal, 1..4, 1..4 );
```

$$A := \text{array}(\mathit{tridiagonal}, 1..4, 1..4, [\,])$$

```
> A[1,2] := x;
```

$$A_{1,2} := x$$

```
> A[1,4] := y;

Error, (in index/tridiagonal)
band width 1 is exceeded, indices = [1,4]
```

```
> print(A);
```

$$\begin{bmatrix} A_{1,1} & x & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 \\ 0 & A_{3,2} & A_{3,3} & A_{3,4} \\ 0 & 0 & A_{4,3} & A_{4,4} \end{bmatrix}$$

```
> A := array( tridiagonal, antisymmetric, 1..4, 1..4 );
```

$$A := \mathrm{array}(\mathit{tridiagonal}, \mathit{antisymmetric}, 1..4, 1..4, [\,])$$

```
> A[1,2] := x;
```

$$A_{1,2} := x$$

```
> print(A);
```

$$\begin{bmatrix} 0 & x & 0 & 0 \\ -x & 0 & A_{2,3} & 0 \\ 0 & -A_{2,3} & 0 & A_{3,4} \\ 0 & 0 & -A_{3,4} & 0 \end{bmatrix}$$

Itt egyszerre alkalmaztuk a `tridiagonal` és az `antisymmetric` indexfüggvényt. Egy elem elérésekor a következő játszódik le:

1. A rendszer először az első nevet veszi, tehát először az **index/tridiagonal** eljárást hívja;
2. ha az index a megadott sávon kívül van, 0-t ad vissza. Egyébként az **index/tridiagonal** a tömb valamely elemét csak akkor fogja elérni, ha a tömb antiszimmetrikus.
3. az **index/antisymmetric** a megfelelő indexezési sorrend szerint kiválaszt egy táblaelemet és ezt visszadja
4. az **index/tridiagonal** ezt az értéket adja vissza

Az indexfüggvényt úgy definiáltuk, hogy magasabb dimenziós tömbökre és ismeretlen értékű tömbelemekre is működjön:

```
> A := array( tridiagonal, 1..3, 1..3, 1..3 );
```

$$A := \mathrm{array}(\mathit{tridiagonal}, 1..3, 1..3, 1..3, [\,])$$

```
> 'A[1,2,3]' = [1,2,3];
```

$$A_{1,2,3} = [1, 2, 3]$$

```
> 'A[1,2,k]' = A[1,2,k];
```

$$A_{1,2,k} = A_{1,2,k}$$

```
> 'A[1,3,k]' = A[1,3,k];
```

$$A_{1,3,k} = 0$$

```
> 'A[1,k,k+1]' = A[1,k,k+1];
```

$$A_{1,\,k,\,k+1} = A_{1,\,k,\,k+1}$$

```
> 'A[1,k,k+2]' = A[1,k,k+2];
```

$$A_{1,\,k,\,k+2} = 0$$

Az *array* típusú adatokra mutató változók kiértékelése különbözik a szokásos *teljes kiértékeléstől*, amelyet például formulákra vagy listákra mutató változók esetében alkalmaz a rendszer. Ez a speciális kiértékelési séma a *table* adatstruktúrára is vonatkozik. A 12.6. alfejezetben részletesen elmagyarázzuk ezen adattípusok kiértékelését annak minden következményével együtt. Most csak egy aspektusra utalunk:

```
> vec := array( [ 1+a, 2+b, 3+c ] );
```

$$vec := [1 + a,\, 2 + b,\, 3 + c]$$

```
> vec;
```

$$vec$$

A vec változó kiértékelése nem mutatja meg a tömböt. Teljes kiértékelést kell kikényszeríteni az **eval**-lal:

```
> eval( vec );
```

$$[1 + a,\, 2 + b,\, 3 + c]$$

Ha az indexhatárokat is látni szeretnénk, használjuk az **op** vagy az **lprint** parancsot:

```
> op( eval(vec) );
```

$$1..3,\, [1 = 1 + a,\, 2 = 2 + b,\, 3 = 3 + c]$$

```
> lprint( eval(vec) );

array(1 .. 3,[(1)=1+a,(2)=2+b,(3)=3+c])
```

12.5. Tábla

A *table* hasonlít más programozási nyelvek (Pascal, Algol68 vagy a C) *rekord* adatstruktúrájára. Az alábbi példában egy *table* típusú objektum látható:

```
> color := table(
> [ red = RGB(1,0,0),
> turquoise = RGB(0.7,0.9,0.9),
> white = RGB(1,1,1)
> ] );
```

$$\begin{aligned}&color := \text{table}([\\ &\quad turquoise = \text{RGB}(.7,\, .9,\, .9)\\ &\quad red = \text{RGB}(1,\, 0,\, 0)\\ &\quad white = \text{RGB}(1,\, 1,\, 1)\\ &\quad])\end{aligned}$$

A tábla bejegyzéseinek belső tárolását a rendszer tördelőtáblázatokban (hash table) végzi; emiatt a tárolás sorrendjét nem tudjuk befolyásolni. A tördelőtáblázatok használata az elemek táblabeli pozíciójától független konstans idejű elérést tesz lehetővé. De nincs is szükségünk a komponensek direkt belső elérésére, mivel a táblák indexei és elemei könnyen visszakereshetők.

```
> indices( color );
```

$$[turquoise], [red], [white]$$

```
> entries( color );
```

$$[\mathrm{RGB}(.7, .9, .9)], [\mathrm{RGB}(1, 0, 0)], [\mathrm{RGB}(1, 1, 1)]$$

A tábla adott elemének kiválasztása vagy megváltoztatása a [] szelekciós operátorral történhet:

```
> color[ red ];
```

$$\mathrm{RGB}(1, 0, 0)$$

```
> color[ red ] := HUE(0);
```

$$color_{red} := \mathrm{HUE}(0)$$

```
> print( color );
```

table([
$turquoise = \mathrm{RGB}(.7, .9, .9)$
$red = \mathrm{HUE}(0)$
$white = \mathrm{RGB}(1, 1, 1)$
])

Új táblastruktúrát indexelt neveknek történő értékadásokkal hozhatunk létre legkönnyebben a Maple-ben. A következő példában lérehozzuk a `Laplace` nevű táblát, amelynek egyetlen bejegyzése az $\mathrm{erf}(\sqrt{t})$ függvény Laplace transzformáltját tartalmazza:

```
> Laplace[ erf(sqrt(t)) ] := 1/(s*sqrt(s+1));
```

$$Laplace_{\mathrm{erf}(\sqrt{t})} := \frac{1}{s\sqrt{s+1}}$$

```
> print( Laplace );
```

table([
$$\mathrm{erf}(\sqrt{t}) = \frac{1}{s\sqrt{s+1}}$$
])

Ha definiálva van az **index/newtable** eljárás, akkor a rendszer minden esetben ezt hívja, amikor olyan indexelt névnek adunk értéket, amelyhez tartozó táblát vagy tömböt korábban még nem definiáltunk. Az eljárás alapföltételezés szerint egy új táblát hoz létre, de tetszés szerint átírható. Például halmazok vagy kifejezéssorozatok elemeinek nem adhatunk értéket `S[x] := y` alakú utasításokkal:

```
> S := {a,b,c}:  Q := a,b,c:
> S[3] := U;

Error, cannot assign to a set

> Q[3] := C;

Error, cannot assign to an expression sequence
```

Az **index/newtable** következő módosított változata már ezt is megengedi:

```
> `index/newtable` := proc( ind, tab, val )
>   if not assigned(tab)
>   then tab := table():  tab[op(ind)] := op(val)
>   elif nops( [ eval(tab) ] ) > 1 # expression sequence
>       and type( ind, [integer] ) and ind[1] >= 1 and
>       ind[1] <= nops( [ eval(tab) ] )
>   then tab := op( subsop( ind[1] = op(val),
>       [ eval(tab) ] ) )
>   elif type( eval(tab), set ) and
>       type( ind, [integer] ) and ind[1] >= 1 and
>       ind[1] <= nops( eval(tab) )
>   then tab := subsop( ind[1]=op(val), eval(tab) )
>   else ERROR(`not a table/array/set/sequence`)
>   fi
> end:
> S; S[3] := U; S;
```

$$\{c, a, b\}$$
$$S_3 := U$$
$$\{c, a, U\}$$

```
> Q; Q[3] := C; Q;
```

$$a, b, c$$
$$Q_3 := C$$
$$a, b, C$$

Folytassuk a táblák vizsgálatát. Nézzük meg, miért a **print**-et használtuk az előző példákban a táblák megjelenítésére, és miért volt szükség olyan sok **eval**-ra az **index/newtable** forráskódjában. Ennek oka az, hogy a *table* típusú adatokra mutató változók kiértékelése különbözik a szokásos *teljes kiértékeléstől.* A 12.6. alfejezetben részletesen elmagyarázzuk a *table* típusú objektumok kiértékelését. Most csak ennek egyik következményére mutatunk rá:

```
> Laplace; # evaluation to the table name
```

$$Laplace$$

Az **op**-ot vagy az **eval**-t használhatjuk, ha magát az adatstruktúrát szeretnénk megjeleníteni:

```
> op( Laplace );
```

table([

$$\mathrm{erf}(\sqrt{t}) = \frac{1}{s\sqrt{s+1}}$$

])

```
> eval( Laplace );
```

table([

$$\mathrm{erf}(\sqrt{t}) = \frac{1}{s\sqrt{s+1}}$$

])

Táblaelem törlése értéktelenítéssel (aposztrófokkal vagy az **evaln**-nel) végezhető el.

```
> color[ red ] := evaln( color[ red ] );
```

$$color_{red} := color_{red}$$

```
> print( color );
```

table([
turquoise = RGB(.7, .9, .9)
white = RGB(1, 1, 1)
])

```
> for c in indices( color ) do
>   color[op(c)] := evaln( color[op(c)] )
> od:
> print( color );
```

table([
])

Most már azt is tudjuk, hogyan hozható létre üres tábla:

```
> table([]);
```

table([
])

Vagy még rövidebben: **table( )**.

Az **array** eljáráshoz hasonlóan a **table**-nek is megadhatók speciális alakú struktúrákra vonatkozó opciók:

```
> distance := table( symmetric );
```

distance := table(*symmetric*, [
])

```
> distance[ Amsterdam, Brussels ] := 157 * km:
> distance[ Amsterdam, `Rio de Janeiro` ] := 9512 * km:
> print( distance );
```

$$\text{table}(symmetric, [\\ (Amsterdam, Brussels) = 157\, km\\ (Amsterdam, Rio\ de\ Janeiro) = 9512\, km\\ ])$$

```
> distance[ Brussels, Amsterdam ];
```

$$157\, km$$

Saját indexfüggvényeket is bevezethetünk az előző alfejezetben megismert módon.

A teljesség kedvéért a 12.4. ábrán megmutatjuk a tömbök és a táblák belső reprezentációját.

table	↑indexing func.	↑array-bounds	↑hash table

12.4. ábra: A táblának megfelelő belső adatsruktúra

12.6. Utolsó név kiértékelés

Az *array*, *table* vagy *procedure* típusú adatokra mutató változók kiértékelése különbözik a szokásos *teljes kiértékeléstől.* Ezt a következő forgatómátrix példájával illusztráljuk:

```
> R := array( [ [ cos(alpha), -sin(alpha) ],
>               [ sin(alpha), cos(alpha) ] ] );
```

$$R := \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

```
> R;
```

$$R$$

```
> whattype( R );
```

$$string$$

```
> type( R, array );
```

$$true$$

Az R változó kiértékelése nem mutatja meg a tömböt. Teljes kiértékelést kell kikényszerítenünk az **eval**-lal:

```
> eval(R);
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

A szekció folytatásából kitűnik, hogy az egyes mátrixelemek valójában még most sem értékelődtek ki:

```
> alpha := 1:  eval(R);
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

```
> R[1,2];
```

$$-\sin(1)$$

```
> map( eval, R );
```

$$\begin{bmatrix} \cos(1) & -\sin(1) \\ \sin(1) & \cos(1) \end{bmatrix}$$

A tömbök, táblák és eljárások kiértékelését *utolsó név kiértékelésnek* (last name evaluation) hívjuk. Ez azt jelenti, hogy a kiértékelés eredménye a kiértékelés közben kapott értékek láncában az az utolsó név, amely közvetlenül megelőzi a lánc végén álló *array*, *table* vagy *procedure* típusú objektumot. A fönti példában tehát a Maple R-t csak saját *string* típusú nevére értékeli ki. A teljes kiértékelés kikényszerítéséhez az **eval**, sőt az egyes elemek kiértékeléséhez még a **map**(**eval**...) is szükséges.

Az utolsó név kiértékelésének hatása természetesen még jobban látszik egy olyan példán, amelyben a változó közvetlenül nem egy tömbre, hanem egy tömb nevére értékelődik ki:

```
> T := S;
```

$$T := S$$

```
> S := R;
```

$$S := R$$

```
> eval(T,1); # value of T
```

$$S$$

```
> eval(T,2); # value of S
```

$$R$$

```
> eval(T,3); # value of R
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

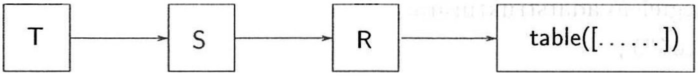
```
> map( eval, T ); # evaluation of matrix entries
```

$$\begin{bmatrix} \cos(1) & -\sin(1) \\ \sin(1) & \cos(1) \end{bmatrix}$$

```
> T; # evaluation of T to last name
```

$$R$$

A belső adatsruktúra a 12.5. ábrán láthatóhoz hasonló.



12.5. ábra: A tömbök kiértékelését ábrázoló belső adatsruktúra

Világos, hogy R, S és T ugyanarra a tömbre mutat. Ha az R[1,2] mátrixelemre hivatkozunk, az eredmény ugyanaz lesz, mintha S[1,2]-t vagy T[1,2]-t választottuk volna.

```
> alpha := 'alpha': # reset alpha to its name
> R[1,2], S[1,2], T[1,2];
```

$$-\sin(\alpha),\ -\sin(\alpha),\ -\sin(\alpha)$$

Ha megváltoztatjuk az S[1,2] mátrixelemet, mindhárom mátrix egyidejűleg ugyanúgy változik. Az utolsó név kiértékelés teszi lehetővé, hogy neveket szinonímákként használhassunk:

```
> eval(R);
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

```
> S[2,1] := 0:
> eval(R), eval(S), eval(T);
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ 0 & \cos(\alpha) \end{bmatrix}, \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ 0 & \cos(\alpha) \end{bmatrix}, \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ 0 & \cos(\alpha) \end{bmatrix}$$

Ha másolatot szeretnénk készíteni egy mátrixról, majd ezt a másolatot úgy szeretnénk megváltoztatni, hogy az eredeti mátrix ne változzon, akkor a **copy** eljárást kell alkalmaznunk:

```
> S := copy(R): S[1,2] := 0:
> eval(R), eval(S);
```

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ 0 & \cos(\alpha) \end{bmatrix}, \begin{bmatrix} \cos(\alpha) & 0 \\ 0 & \cos(\alpha) \end{bmatrix}$$

12.7. Függvényhívások

A függvényhívásokat rekord adatstruktúraként használhatjuk. A polárkoordináta-rendszerben fölírt komplex számokat például **polar**(*absolut value, argument*) alakú függvényhívásokkal reprezentálja a Maple:

```
> polar(3,Pi/2);
```

$$\mathrm{polar}(3, \frac{1}{2}\pi)$$

Itt a `polar(...)` csak helykitöltőként szerepel, hogy más eljárások hivatkozhassanak a megfelelő adatstruktúrára:

```
> evalc(");
```

$$3\,I$$

A függvényhívások rekord adatstruktúraként való használatára további példákat nyújt a **RootOf**, **DESol** és a **PIECEWISE**.

A függvényhívások egyik kellemes aspektusa, hogy a hívás kiírási módja általunk is megadható. Polárkoordinátákkal fölírt komplex számokra definiálhatnánk a következő nyomtató rutint:

```
> `print/polar` := proc(r,a) r*e^(I*a) end:
> polar(5,Pi/3); # pretty display
```

$$5\,e^{(1/3\,I\,\pi)}$$

```
> whattype("); # still a function call
```

function

```
> lprint(""); # internal representation

polar(5,1/3*Pi)

> evalc("");
```

$$\frac{5}{2} + \frac{5}{2}\,I\,\sqrt{3}$$

```
> convert(",'polar');
```

$$5\,e^{(1/3\,I\,\pi)}$$

A függvényhívások komponenseit az **op**-pal választhatjuk ki:

```
> op(2,");
```

$$\frac{1}{3}\pi$$

A **macro** lehetőségeit fölhasználva értelmes neveket adhatunk a szelekciós operátoroknak:

```
> macro( radius=1, argument=2 ):
> op( radius, "" ), op( argument, "" );
```

$$5,\ \frac{1}{3}\pi$$

Természtesen definiálhatunk a kiválasztást végző eljárásokat is:

```
> macro( radius = radius ):
> radius := proc(z)
>   if type( z, polar(anything,anything) )
>   then op(1,z)
>   else 'procname(args)'
>   fi
> end:
> polar(5,Pi/3);
```

$$5\,e^{(1/3\,I\,\pi)}$$

```
> radius(");
```

$$5$$

A függvényhívás valamely komponense a **subsop**-pal változtatható meg:

```
> subsop( argument = Pi/6, "" );
```

$$5\,e^{(1/6\,I\,\pi)}$$

Függvényhívásokra is könnyen kiterjeszthetők az **evalf**, **simplify**, **convert** stb. könyvtári rutinok. A komplex számok lebegőpontos kiértékelését elvégző kiterjesztés például a következő lehet:

```
> `evalf/polar` := proc()
>   local z;
>   z := polar( evalf(args[1]), evalf(args[2]) );
>   if type( z, polar(float,float) )
>   then op(1,z) * exp( op(2,z)*I )
>   fi
> end:
> polar(5,Pi/3);
```

$$5\,e^{(1/3\,I\,\pi)}$$

```
> evalf(");
```

$$2.500000001 + 4.330127019\,I$$

Amikor az **evalf**-et olyan kifejezésre kell alkalmazni, amely **polar**(...) alakú függvényhívást tartalmaz, a Maple az **evalf/polar** eljárás definícióját keresi meg és (szükség esetén) tölti be a Maple könyvtárból vagy a fölhasználó saját könyvtárából. Fölhasználhatjuk az előző szekció eredményeként rendelkezésünkre álló **evalf/polar** definíciót is.

A Maple lehetőségeinek kiterjesztésére vonatkozó második példaként a polárkoordinátákról Descartes-féle koordinátákra való konverziót tekintjük:

```
> `convert/cartesian` := proc(z)
>   local r,a;
>   if type( z, polar(anything,anything) )
>   then r := op(1,z); a := op(2,z);
>     r*cos(a) + r*I*sin(a)
>   else 'procname(args)'
>   fi
> end:
> polar(5,Pi/3);
```

$$5\,e^{(1/3\,I\,\pi)}$$

```
> convert( ", cartesian );
```

$$\frac{5}{2} + \frac{5}{2} I \sqrt{3}$$

```
> polar(R,phi): " = convert( ", cartesian );
```

$$R\, e^{(I\, \phi)} = R \cos(\phi) + I\, R \sin(\phi)$$

Valahányszor a **convert**(..., *type*) eljárást hívjuk, a Maple a **convert/type** eljárás definícióját keresi meg és (szükség esetén) tölti be a Maple könyvtárból vagy a fölhasználó saját könyvtárából. Az előző szekció eredményeként rendelkezésünkre áll a **convert/cartesian** definíciója, ezt alkalmazta a kifejezésre.

12.8. Összetett adattípusok közti konverziók

Már láttuk, hogyan alakítható át sorozat listává vagy halmazzá: csupán kapcsos, illetve szögletes zárójelbe kell tenni az adott objektumot. A *set* vagy a *list* típusú objektumok az **op** függvény segítségével konvertálhatók. A Maple halmazokra és listákra alkalmazható explicit konverziós rutinokat is rendelkezésünkre bocsát. Az alábbiakban néhány olyan példa következik, amely ezeket a konverziókat összegzi:

```
> mySequence := a, b, b, q;
```

$$mySequence := a, b, b, q$$

```
> myList := [ mySequence ]; # sequence -> list
```

$$myList := [a, b, b, q]$$

```
> mySet := { mySequence }; # sequence -> set
```

$$mySet := \{a, b, q\}$$

```
> op( myList ); # list -> sequence
```

$$a, b, b, q$$

```
> myList[]; # list -> sequence
```

$$a, b, b, q$$

```
> op( mySet ); # set -> sequence
```

$$a, b, q$$

```
> mySet[]; # set -> sequence
```

$$a, b, q$$

```
> convert( myList, set ); # list -> set
```

$$\{a, b, q\}$$

```
> convert( mySet, list ); # set -> list
```

$$[a, b, q]$$

Az ilyen explicit konverziós rutinok használhatók egyrészt halmazok és listák, másrészt tömbök és függvényhívások közti konverzióra is:

```
> myArray := array( myList ); # list -> array
```

$$myArray := [a, b, b, q]$$

```
> convert( myList, array ); # list -> array
```

$$[a, b, b, q]$$

```
> convert( myArray, list ); # array -> list
```

$$[a, b, b, q]$$

```
> convert( myArray, set ); # array -> set
```

$$\{a, b, q\}$$

```
> Call := F( mySequence ); # sequence -> function call
```

$$Call := \mathrm{F}(a, b, b, q)$$

```
> convert( Call, list ); # function -> list
```

$$[a, b, b, q]$$

```
> convert( Call, set ); # function -> set
```

$$\{a, b, q\}$$

```
> myArray2D := array( [ myList, myList ] );
```

$$myArray2D := \begin{bmatrix} a & b & b & q \\ a & b & b & q \end{bmatrix}$$

```
> convert( myArray2D, listlist ); # array -> list of lists
```

$$[[a, b, b, q], [a, b, b, q]]$$

```
> convert( ", array ); # list of lists -> array
```

$$\begin{bmatrix} a & b & b & q \\ a & b & b & q \end{bmatrix}$$

Ezek a konverziók magasabb dimenziójú tömbök esetében is működnek. Az előző konverziók kombinációit listák alakjának megváltoztatására is fölhasználhatjuk:

```
> L := [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 ]:
> matrix( 2,6, L ); # array
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

```
> convert( ", listlist ); # array -> list
```

$$[[1, 2, 3, 4, 5, 6], [7, 8, 9, 10, 11, 12]]$$

```
> convert( matrix( 6,2,L ),listlist ):
> setattribute( ", matrix ); # display in matrix format
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

```
> op(");
```

$$[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12]$$

```
> map(op,"");
```

$$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$$

A 12.3. táblázatban az összetett adattípusok közti fontosabb konverziókat soroljuk föl.

S\T	**seq.**	**set**	**list**	**array**	**call**
seq.		{"}	["]	array(["])	f(")
set	op(")		[op(")]	array([op(")])	f(op("))
	"[]		convert(“, list)		f("[])
list	op(")	{op(")}		array(")	f(op("))
	"[]	convert(", set)		convert(", array)	f("[])
array		convert(", set)	convert(“, list)		
call	op(")	convert(", set)	convert(“, list)		

12.3. táblázat: Az S és a T összetett adattípusok közti konverziók

12.9. Gyakorlatok

1. Legyen U az első 10 prímszám halmaza. Legyen V az első 20 darab $2^n - 1$ alakú természetes szám halmaza.

   (a) Állítsuk elő az U és a V halmazokat a Maple rendszerben.

   (b) Számítsuk ki az $U \cup V$ és az $U \cap V$ halmazokat.

2. (a) Generáljunk 100 darab -10 és 10 közé eső véletlen számból álló listát.

   (b) Hagyjuk el a listából a többszörösen előforduló elemeket.

   (c) Válasszuk ki a (b) feladat eredményeként kapott listából azokat a számokat, amelyek 2-vel vagy 3-mal oszthatók.

(d) Válasszuk ki a (b) feladat eredményeként létrejövő listából az 5-nél nagyobb számokat.

(e) Határozzuk meg, hogy a (d) feladat eredményeként kapott listában szereplő számok hányszor fordulnek elő az eredeti listában.

3. Természetesen tudjuk, hogy az $x^{50} - x^{20} = 0$ egyenletnek hány megoldása van a komplex számok fölött. Amennyiben a **solve** eljárást használjuk egyenletek megoldására, a képernyőn megjelenő outputban egymás után sorakozó egyenlőségek halmaza alapján nehezen számolhatjuk meg a Maple által talált megoldásokat. Határozzuk meg a **nops** és a **map** eljárások segítségével a fenti egyenlet megoldásainak a számát. Ellenőrizzük a választ visszahelyettesítéssel. (Egy jó tanács: annak érdekében, hogy használni tudjuk a **map** és a **subs** kombinációját, tekintsük az egyenletet egy egyenletből álló egyismeretlenes egyenletrendszernek.)

4. Hogyan duplázhatók meg az $[a, b, c]$ lista elemei úgy, hogy az $[a, a, b, b, c, c]$ listát kapjuk? Hogyan alakíthatjuk át ezt $[[a, a], [b, b], [c, c]]$ alakúra? Alkalmazhatók-e ezek a megoldások az első 10 000 természetes számból álló lista elfogadható időn belül végrehajtható hasonló átalakítására?

5. Írjunk olyan `maketable` eljárást, amely az indexek és az elemek listája alapján létrehozza a megfelelő táblát. A `maketable([red,green],[0,1/3])` hívás hatására például a `red` $\to$ 0, `green` $\to \frac{1}{3}$ hozzárendeléseknek megfelelő táblázatot kell generálnia.

6. Hozzuk létre az

$$M_{ij} = x^{\gcd(i+1,j+1)} + 1$$

képlettel definiált 3×3-as M mátrixot. Írjuk föl faktorizált alakú elemekkel is.

7. Hozzuk létre az

$$M_{ij} = 10^{-(i^2+j^2)}$$

képlettel definiált 5×5-ös M mátrixot, és a kicsi mátrixelemeket az **fnormal** eljárással normalizáljuk 0-ra.

8. Hozzuk létre azt a 7×7-es alsó trianguláris M mátrixot, amelynek minden nemzérus eleme 1. Konvertáljuk a sorfolytonos rendezésnek megfelelően egészek listájává.

9. Általánosítsuk a 12.4. alfejezetben szereplő `tridiagonal` indexfüggvényt a `width` ≥ 1 sávszélességű szalagmátrixokra.

10. Definiáljunk a fölső trianguláris mátrixoknak megfelelő speciális indexfüggvényt.

11. Definiáljunk olyan `polynomial` nevű speciális indexfüggvényt, ami a megfelelő tömbökben csak polinomokat enged meg elemekként.

12. Reprezentáljuk az a és b racionális számokkal fölírt $a + b\sqrt{2}$ alakú algebrai számokat rendezett párokkal.

(a) Implementáljuk ezt a számot a Maple-ben az $[a, b]$ listával. Írjuk meg az `ADD`, `MUL` és a `DIV` eljárásokat, amelyek az így definiált algebrai számok összegét, szorzatát, illetve hányadosát kanonikus alakban állítják elő.

(b) Implementáljuk a fönti algebrai számokat `algnum(a,b)` alakú függvényhívásokkal. Írjuk meg azt a `print/algnum` eljárást, amely a számot a szokásos matematikai jelöléseknek megfelelő $a + b\sqrt{2}$ alakban írja ki.

(c) A (b) rész implementációjának megfelelően írjuk át az `ADD`, `MUL` és a `DIV` eljárásokat.

13.

Az **assume** parancs

Az **assume** parancs változók és kifejezések tulajdonságainak specifikálását teszi lehetővé. Ez a fejezet bevezetésül szolgál; a mögötte álló indítékokat, az alkalmazott modellt és ennek Maple implementációját írja le. A [49] irodalom egy informális bevezető az **assume**-ról, [189, 190, 191] részletesebben tárgyalja az alapul szolgáló modellt.

13.1. Miért van szükség az **assume**-ra?

Az **assume** parancs bevezetését a Maple-ben főleg az motiválta, hogy így lehetővé válik a változók értelmezési tartományáról való ismereteink beépítése, s ezzel az egyszerűsítések minél teljesebb végrehajtása az alkalmazott transzormációk érvényességének veszélyeztetése nélkül.

```
> expr := sqrt(x^2);
```

$$expr := \sqrt{x^2}$$

Ezt a kifejezést a Maple nem egyszerűsíti automatikusan x-re vagy abs(x)-re. Általában mindkét megoldás helytelen. A **simplify** parancs a négyzetgyök függvényt többértékű komplex függvénynek tekinti, s ez néha meglepő következményekkel jár.

```
> simplify( expr );
```

$$\mathrm{csgn}(x)\, x$$

A kifejezést az x komplex előjelfüggvényét tartalmazó alakra transzformálta a rendszer. Ha közelebbről meghatározzuk számításaink tartományát vagy x értékét, a Maple is konkrétabb választ ad:

```
> assume( x, real ):
> simplify( expr );
```

$$\operatorname{signum}(x^{\sim})\, x^{\sim}$$

Vagy ha inkább az abszolút érték függvényt preferáljuk:

```
> convert( ", abs );
```

$$|x^{\sim}|$$

```
> assume( x>0 ):
> simplify( expr );
```

$$x^{\sim}$$

```
> assume( x<0 ):
> simplify( expr );
```

$$-x^{\sim}$$

Az x után írt tilde jelzi, hogy valamilyen rá vonatkozó föltevést tettünk.

A következő két példa azt mutatja, hogy több információ birtokában a Maple jobb eredmények elérésére képes:

```
> integral := int( sin(a*x)/x , x=0..infinity );
```

$$\mathit{integral} := \frac{1}{2}\operatorname{signum}(a)\,\pi$$

```
> assume( a<0 ):
> integral;
```

$$-\frac{1}{2}\pi$$

```
> integral := integrate( exp(-s*t), t=0..infinity );
```

$$\mathit{integral} := \lim_{t\to\infty} -\frac{e^{(-s\,t)}}{s} + \frac{1}{s}$$

Ez a részeredmény azt mutatja, hogy a Maple szerint a határérték meghatározása könnyebb probléma, mint az integrálás. Ha többet árulunk el s-ről, ki is tudja számolni a határértéket a Maple:

```
> assume( s>0 ):
> integral;
```

$$\frac{1}{s^{\sim}}$$

```
> assume( s<=0 ):
> integral;
```

$$\infty$$

A Maple segítségével tehát megtaláltuk az

$$\int_0^\infty e^{-st}\,dt = \begin{cases} \dfrac{1}{s} & \text{ha } s > 0 \\ \infty & \text{egyébként} \end{cases}$$

formulát. A Maple számolás közben nemcsak a változók előjelét vizsgálja; ennél többet végez. Ha például definiáljuk a `c` valós változót, és `s`-nek `c` négyzetét adjuk értékül, akkor a Maple már ebből tudni fogja, hogy a fönti integrál melyik ágát válassza:

```
> assume( c, real ):
> s := c^2;
```

$$s := c^{\sim 2}$$

Figyeljük meg alaposan: `c~`-t emeltük négyzetre és nem `c`-t a ~2 hatványra! Még egy tudnivaló: az értékadás hatására a Maple elfelejti az összes `s`-re vonatkozó korábbi kikötést:

```
> integral;
```

$$\frac{1}{c^{\sim 2}}$$

A fönti példák tehát azt támasztják alá, hogy pótlólagos kikötéseket kell tennünk, ha a Maple által visszaadott eredmény

- valamely integrálra vonatkozó kiértékeletlen **limit**;
- nincs az általunk várt vagy kért formára egyszerűsítve;
- kiértékeletlen **csgn**, **signum** vagy **abs** függvényhívás;
- kiértékeletlen **Re** vagy **Im** függvényhívás.

Egy jótanács:

Szükséges kikötéseinket a számolás során a lehető legkorábban adjuk meg.

Ezzel elkerülhetjük, hogy a Maple egy általános probléma megoldását utólag specializálja a föltételeinknek megfelelően, s így esetleg egyszerűbb, de hibás eredményt kapjon. Ezt a nézőpontot támogatja a [3]-ban fölvetett következő kérdés vizsgálata: a hengerkoordinátákkal fölírt alábbi

$$V = \frac{1}{8\pi\epsilon_0}\left(\frac{1}{\sqrt{r^2+(z-i)^2}} + \frac{1}{\sqrt{r^2+(z+i)^2}}\right)$$

komplex függvény tekinthető-e valamely fizikai jelentéssel bíró töltéssűrűség által generált elektrosztatikus potenciálnak. A $z = 0$ síkban a töltéssűrűséget a

$$\sigma = \epsilon_0\left(\lim_{x\downarrow 0}\frac{\partial V}{\partial z} - \lim_{x\uparrow 0}\frac{\partial V}{\partial z}\right)$$

formula határozza meg. A cikk szerint a Maple és más számítógépes algebrai rendszerek a $\sigma = 0$ hibás eredményt adják. Az alábbi szekció azt mutatja meg, hogy r-re megfelelő kikötéseket téve a Maple hogyan tudja mégis megtalálni a helyes választ:

```
> V := 1/(8*Pi*epsilon[0]) * ( 1/sqrt(r^2+(z-I)^2) +
>   1/sqrt(r^2+(z+I)^2) );
```

$$V := \frac{1}{8}\,\frac{\dfrac{1}{\sqrt{r^2+(z-I)^2}}+\dfrac{1}{\sqrt{r^2+(z+I)^2}}}{\pi\,\varepsilon_0}$$

```
> E[z] := - diff( V, z );
```

$$E_z := -\frac{1}{8}\,\frac{-\dfrac{1}{2}\,\dfrac{2z-2I}{(r^2+(z-I)^2)^{3/2}}-\dfrac{1}{2}\,\dfrac{2z+2I}{(r^2+(z+I)^2)^{3/2}}}{\pi\,\varepsilon_0}$$

```
> sigma := epsilon[0] * ( limit(E[z], z=0, right) -
>   limit( E[z], z=0, left) );
```

$$\sigma := 0$$

```
> assume( r>=1 );
> sigma := epsilon[0]*(
> limit( E[z], z=0, right) - limit( E[z], z=0, left) );
```

$$\sigma := 0$$

```
> assume( 0<r, r<1 );
> sigma := epsilon[0] * ( limit(E[z], z=0, right) -
> limit( E[z], z=0, left) );
```

$$\sigma := -\frac{1}{2}\,\frac{1}{\pi\,(1-r\tilde{}^2)^{3/2}}$$

Most nézzük meg, mi történik, ha az utolsó kifejezést az $r = 1/2$ helyen vesszük:

```
> r := 1/2: "";
```

$$-\frac{1}{2}\,\frac{1}{\pi\,(1-r\tilde{}^2)^{3/2}}$$

Miért tartalmazza még mindig `r~`-t a válasz? Ennek oka a következő: az `assume( 0<r, r<1)` paranccsal az `r`-t egy olyan új `r~` változóval helyettesítettük, amely a $(0,1)$-ből vehet föl értékeket. Ettől kezdve az `r~` változóra az `r`-rel hivatkozhatunk. Ezután az új változó függvényében meghatároztuk a töltéssűrűséget. De most nem az `r~`-nak, hanem az `r`-nek adtuk értékül az 1/2-et. A töltéssűrűség azonban továbbra is `r~`-t tartalmazza. Az `r:=1/2` értékadással tehát megszűnt annak lehetősége, hogy egyszerű módon hivatkozhassunk `r~`-re. A 3.6. alfejezetben részletesebben elmagyaráztuk az **assume** implementálásának ezt a mellékhatását, és azt is leírtuk, hogyan adható érték olyan változóknak, amelyekre valamilyen kikötéseket tettünk.

13.2. Az **assume** alapjai

Az **assume**-mal kapcsolatos lehetőségek a 13.1. táblázatban fölsorolt öt alapparanccsal érhetők el. A hatodik utasítással a Maple tudását bővíthetjük új tulajdonság definiálásával.

Parancs	**Jelentése**
assume	föltétel(ek) megadása
additionally	kiegészítő föltétel(ek) megadása
about	a most érvényes föltétel(ek) kiírása
is	tulajdonság lekérdezése
coulditbe	tulajdonság lehetőségének meghatározása
addproperty	tulajdonság hozzáadása a Maple tudásbázisához

13.1. táblázat: Az **assume**-mal kapcsolatos parancsok

Az **assume** eljárással a változók tulajdonságait és a változók közti kapcsolatokat adhatjuk meg a Maple-nek. Alapvetően két formája van:

assume(< *inequality* >)

és

assume(< *variable* > , < *property* >).

Ha egy kifejezésre vonatkozó föltételeket adunk meg, a továbbiakban a kifejezésben szereplő változók neve után egy tilde karaktert tesz a Maple, ezzel jelezve, hogy rájuk bizonyos kikötések érvényesek. Az **about** eljárás a változókra és kifejezésekre pillanatnyilag érvényes föltételeket jeleníti meg.

Például azt, hogy r egynél kisebb abszolút értékű valós konstans, többféle módon megadhatjuk:

```
> assume( abs(r)<1 );
> about( r );

Originally r, renamed r~:
  Involved in the following expressions with properties
    abs('r~') assumed RealRange(-infinity,Open(1))
  also used in the following assumed objects
  [abs('r~')] assumed RealRange(-infinity,Open(1))

> assume( r, RealRange( Open(-1), Open(1) ) );
> about( r );

Originally r, renamed r~:
  is assumed to be: RealRange(Open(-1),Open(1))

> assume( -1<r, r<1 ):
> about( r );

Originally r, renamed r~:
  is assumed to be: RealRange(Open(-1),Open(1))
```

Az **about** eredményeit is megadtuk, így látható, hogy a különféle föltételeket nem mindig ugyanúgy kezeli a rendszer.

Az utolsó előtti parancs azt mutatja, hogy az **assume** egy hívásakor több föltételt is összekombinálhatunk. Figyeljük meg a különbséget:

```
> assume( -1<r ):
> assume( r<1 ):
> about( r );

Originally r, renamed r~:
  is assumed to be: RealRange(-infinity,Open(1))
```

Valahányszor az **assume**-mal föltételezünk valamit, a föltételben szereplő változókra vonatkozó összes korábbi kikötés törlődik. Legalábbis ez a helyzet, ha az `_Envadditionally` környezeti változó értéke az alapértelmezés szerinti `false`:

```
> _Envadditionally := true:
> assume( -1<r ):
> about( r );

Originally r, renamed r~:
  is assumed to be: RealRange(Open(-1),Open(1))
```

Ahelyett, hogy az `_Envadditionally`-nak a `true` értéket adnánk, használhatjuk az **additionally** eljárást is.

```
> additionally( arcsin(r),
>   RealRange( Open(-Pi/2), Open(Pi/2) ) );
> about( r );

Originally r, renamed r~:
  Involved in the following expressions with properties
    arcsin('r~') assumed RealRange(Open(-1/2*Pi),Open(1/2*Pi))
  is assumed to be: RealRange(Open(-1),Open(1))
  also used in the following assumed objects
  [arcsin('r~')] assumed RealRange(Open(-1/2*Pi),Open(1/2*Pi))
```

A következő könyvtári függvények veszik figyelembe a megadott föltételeket:

csgn, **signum**, **abs**, **Re**, **Im**, **frac**, **trunc**, **round**, **ceil**, **floor**.

Az `r`-re tett fönti kikötések után a Maple már önállóan is boldogul:

```
> abs( r - 1 ), signum( r^5 - 1 );
```

$$1 - r^{\sim}, -1$$

A **coulditbe** paranccsal azt is megkérdezhetjük, hogy vajon valamely tulajdonság ellentmond-e az eddigi kikötéseknek:

```
> coulditbe( r>-3 ); # no conflict
```

true

```
> coulditbe( r>1/3 ); # also no conflict
```

true

A föntebb fölsorolt függvények az ismert tulajdonságok valamelyikének meglétére az **is** eljárással kérdeznek rá. Ezt mi is használhatjuk tulajdonságok tesztelésére. Az `r`-re kirótt korábbi föltételek esetén

```
> is( r > 1/3 ); # not necessarily true
```

false

```
> is( (1-r^2)/(1+r^2) <= 1 );
```

true

Az **is** eredménye `true`, `false` vagy `FAIL`. Az **is** által visszaadott érték akkor `FAIL`, ha nem tudta eldönteni, hogy a szóbanforgó tulajdonság igaz vagy hamis. Ezt okozhatja az, hogy kevés volt a rendelkezésére álló információ, vagy nem volt képes a szükséges logikai következtetések levezetésére, esetleg nem használt föl elegendően sok erőforrást annak eldöntésére, hogy az adott tulajdonség igaz, hamis vagy eldönthetetlen. Az **is** által visszadott `true` érték azt jelenti, hogy a kifejezés minden lehetséges értéke rendelkezik a kívánt tulajdonsággal. Ha az **is** a `false` eredménnyel tér vissza, akkor van legalább egy olyan lehetséges értéke a kifejezésnek, amelyre nem teljesül az adott tulajdonság.

A következő két alfejezetben a tulajdonságok algebráját és ennek implementációját vizsgáljuk a Maple rendszerben. Ezek után bizonyára az Olvasó is képet tud alkotni arról, hogy mire képes a Maple a tulajdonságokkal kapcsolatban. Most csak annyit jegyzünk meg, hogy az **is** az _EnvTry környezeti változó értékétől függően két módon működhet: a `normal` és a `hard` módon.

```
> _EnvTry := normal:  # the default value
> assume( a<=b, b<=c, c<=a );
> is( a = c ); # successful equality test
```

true

```
> seq( x.i >= x.(i-1), i=1..10 );
```

$$x0 \le x1,\ x1 \le x2,\ x2 \le x3,\ x3 \le x4,\ x4 \le x5,\ x5 \le x6,\ x6 \le x7$$
$$x7 \le x8, x8 \le x9,\ x9 \le x10$$

```
> assume( ", x10 <= x0 ):
> is( x0 = x10 ); # test failure
```

FAIL

```
> _EnvTry := hard:  # let Maple do its very best
> is( x1 = x10 ); # successful equality test
```

true

A fönti példa kapcsán két további technikai kérdést vizsgálunk meg.

- Az **assume** indexelt változókra nem alkalmazható, mivel ez gondokat okozna többek között az index helyettesítését végző parancsoknál vagy a ciklusváltozóval megegyező értékű indexeknél.

- Az is eljárás is használja a remember opciót. Vagyis ha egy teszt eredményeként FAIL vagy valamely más eredményt kaptunk, onnantól kezdve az lesz az eredmény, függetlenül attól, hogy az _EnvTry-t a hard értékre állítottuk-e. Most már világos, hogy a változó hatását demonstráló előző Maple szekcióban miért nem kérdeztünk rá másodszor is az $x0 = x10$ egyenlőségre, miért vettünk helyette egy másikat.

A második megszorítással kapcsolatos problémák elkerülésére létezik olyan alternatív megoldás is, amely minden remember opciót használó könyvtári függvény esetében alkalmazható.

Ha a Maple az újraszámolás helyett egy előzőleg meghatározott értéket keres vissza, explicit módon kérhetjük, hogy felejtse el az emlékezőtáblájában tárolt korábbi értékeket.

Az előző példában tehát így járhatunk el:

```
> readlib( forget ): # load library function
> forget( is ): # forget results of "is"
> _EnvTry := normal: # reset assume mode
> is( x1 = x10 ); # unsuccessful equality test
```

$$FAIL$$

13.3. A tulajdonságok algebrája

A tulajdonságokkal kapcsolatos probléma például

Adottak:	az m és n páratlan egész számok
Kérdés :	igaz-e, hogy $m^2 + n$ páratlan egész?

vagy még általánosabban

Adottak:	$obj_1 \in Prop_1, \ldots, obj_n \in Prop_n$
Kérdés :	igaz-e, hogy $f(obj_1, \ldots, obj_n) \in Prop_0$?

Ilyen kérdések az objektumok tulajdonságainak manipulálásával válaszolhatók meg. Vegyük a fönti példát: „adottak az m és n páratlan egész számok, a kérdés: igaz-e, hogy $m^2 + n$ páratlan egész?”. A probléma két részre osztható:

1. Transzformáljuk a tulajdonságokat az objektumokkal együtt: például ha m páratlan egész, akkor m^2 is páratlan egész. A négyzet függvényhez tehát a tulajdonságokon értelmezett olyan függvény van rendelve, amely a „páratlan egész” tulajdonságot saját magára képezi le. Ha tovább haladunk az $(m, n) \mapsto m^2 + n$ példa vizsgálatával, a következő leképezéseket kapjuk:

$$\begin{aligned} (m, n) &\mapsto add \circ (sqr, id)(m, n) \\ (\mathcal{P}, \mathcal{P}) &\mapsto \overline{add \circ (sqr, id)}(\mathcal{P}, \mathcal{P}) \end{aligned}$$

Itt *add*, *sqr* és *id* az összeadást, a négyzetreemelést, illetve az identikus függvényt jelöli, a $\mathcal{P}$ tulajdonság jelentése „páratlan egész”.

2. A feladatot redukáljuk az

$$\overline{add \circ (sqr, id)}(\mathcal{P}, \mathcal{P}) = \overline{add} \circ (\overline{sqr}, \overline{id})(\mathcal{P}, \mathcal{P}) \subseteq \mathcal{P}?$$

kérdés eldöntésére.

Általánosabban fogalmazva a két lépés a következő:

1. Transzformáljuk a tulajdonságokat az objektumokkal együtt:

$$(obj_1, \ldots, obj_n) \mapsto f(obj_1, \ldots, obj_n)$$
$$(Prop_1, \ldots, Prop_n) \mapsto \overline{f}(Prop_1, \ldots, Prop_n)$$

2. Redukáljuk a feladatot az

$$\overline{f}(Prop_1, \ldots, Prop_n) \sqsubseteq Prop_0$$

kérdés eldöntésére, ahol az $\sqsubseteq$ olyan objektumokon értelmezett reláció, amelyre

$$obj \in Prop \Rightarrow obj \in Prop' \quad \text{iff} \quad Prop \sqsubseteq Prop'.$$

Mivel a tulajdonságokat azonosítjuk az összes olyan objektum halmazával, amelyekre teljesül az adott tulajdonság, a $\sqsubseteq$ reláció a halmazelméleti tartalmazás lesz.

Példák a Maple-ben értelmezett tulajdonságokra:

`integer`	az egészek halmaza
`odd`	a páratlan egészek halmaza
`RealRange(0,Open(1))`	$[0, 1) = \{x \in \mathbb{R} \mid 0 \leq x < 1\}$
`monotonic`	a monoton valós függvények halmaza
`InfinetelyDifferentiable`	a C^∞ függvények halmaza
`Non(singular)`	a nemszinguláris mátrixok halmaza

A tulajdonságok a halmazelméleti tartalmazásra nézve hierarchiákat alkotnak.

$$\texttt{posint} \subset \texttt{natural} \subset \texttt{integer}.$$

Az `integer`, `fraction` stb. tulajdonságok és a `0`, `1` stb. konstansok alkotják az alaptulajdonságokat. Ezekből a $\neg$, $\wedge$ és a $\vee$ indukált hálóműveletekkel képezhetünk újabb tulajdonságokat. A műveletek jelentése:

$\neg$ logikai tagadás (not), például „nemnulla” (a Maple jelölése `Non(0)`)

$\wedge$ logikai és (and), például „egész és kettőnél nagyobb” (a Maple jelölése `AndProp( integer, RealRange(2,infinity) )`).

- $\vee$ logikai vagy (or), például „egyváltozós vagy kétváltozós" (a Maple jelölése `OrProp( unary, binary )`).

A $\top$ legnagyobb hálóelem jelenti a „létező", „nincs semmi kikötés" tulajdonságot. Maple jelölése `TopProp`. Ennek szimmetrikus párja a $\perp$ legkisebb elem, ami a „nem létező" tulajdonságnak felel meg, vagyis olyan kikötés, amelyet teljesítő Maple objektum nem létezhet. Maple jelölése `BottomProp`.

Az összetett függvényekkel kapcsolatos kérdéseket a következő módszerrel kezeli a rendszer. Az objektumokon értelmezett minden f függvénynek megfelel egy $\overline{f}$ tulajdonságokon értelmezett függvény. Például a $+$ addíciós operátor azt a tulajdonságokon értelmezett $\overline{+}$ operátort indukálja, amelyre

$$\begin{aligned}
\overline{+}(\texttt{odd},\texttt{odd}) &= \texttt{even}\\
\overline{+}(\texttt{odd},\texttt{even}) &= \texttt{odd}\\
\overline{+}(\texttt{even},\texttt{odd}) &= \texttt{odd}\\
\overline{+}(\texttt{even},\texttt{even}) &= \texttt{even}\\
\overline{+}(\texttt{prime},\texttt{composite}) &= \texttt{integer}\\
\overline{+}(\texttt{fraction},\texttt{fraction}) &= \texttt{rational}\\
\overline{+}(\texttt{irrational},\texttt{irrational}) &= \texttt{real}
\end{aligned}$$

És így tovább. Természetesen nem szükséges, hogy $\overline{f}$ minden tulajdonságra definiálva legyen. Ha modellünkhöz hozzávesszünk néhány alapvető tulajdonságfüggvényt, például a $\overline{+}, \overline{*}$, $\overline{\ln}$ és az $\overline{\exp}$-t, valamint ezek inverzeit, a tulajdonságok olyan algebráját kapjuk, amely kielégíti a klasszikus hálóaxiómákat és még néhány olyan extra axiómát, amelyekkel fölépíthető a tulajdonságok effektív kalkulusa. A tulajdonságok ténylegesen egy Boole-algebrát alkotnak.

13.4. Az **assume** implementálása

A Maple jelenlegi változata által elfogadott tulajdonságok a következők:

(a) *tulajdonságnevek*, amelyek így osztályozhatók:

alias, például `AndProp( real, constant)` helyett `realcons`;

numerical, például `imaginary` vagy `prime`;

matricial, például `SquareMatrix` vagy `PositiveDefinite`;

functional, például `unary` vagy `OddMap`;

egyéb, például `TopProp`, `MutuallyExclusive`.

(b) *a legtöbb típus*, például `integer`, `fraction` és `rational`. Ide tartoznak a konstansok, így a `0` és az `1` is.

(c) *numerikus tartományok*, például `RealRange( -infinity, Open(0) )`.

(d) *tulajdonságok konjunkciója*, például `AndProp( integer, positive )`.

(e) *tulajdonságok diszjunkciója,* például `OrProp( positive, negative )`.

(f) *tulajdonságok tartománya,* például `PropRange( Prop1, Prop2 )`. Ha `P = PropRange( Prop1, Prop2 )`, akkor minden `Prop1` tulajdonságú objektum rendelkezik a `P` tulajdonsággal is, továbbá minden `P` tulajdonságú objektumnak megvan a `Prop2` tulajdonsága is.

(g) *lineáris tulajdonságok,* például `LinearProp( 3, integer, 0 )` a hárommal osztható egészek halmazát reprezentálja.

(h) *parametrikus tulajdonságok,* például `Non(0)` vagy `Non(singular)`.

A Maple-ben használható tulajdonságok teljes listáját az on-line help rendszer **?property** parancsával kaphatjuk meg. A manuáloldal fölépítése is azt tükrözi, hogy a fönti (a) és (b) osztályba sorolt alaptulajdonságok irányított aciklikus gráfot alkotnak. A fejezet végén található 13.1., 13.2. és 13.3. ábra az előre definiált numerical, matricial, illetve functional tulajdonságok gráfját ábrázolja.

Új tulajdonság megadásához szükséges összes szülőjének és leszármazottjának fölsorolása a `‘property/ParentTable‘` és a `‘property/ChildTable‘` táblákban. Valamely alaptulajdonságnak az irányított aciklikus gráfban elfoglalt pozícióját az **about**-tal tudhatjuk meg, például

```
> about( integer );

integer:
  a known property having {rational, GaussianInteger} as
  immediate parents and {1, composite, prime} as immediate
  children. mutually exclusive with {fraction, irrational}
```

Az **addproperty** eljárással installálhatunk új tulajdonságot a tulajdonságok hierarchiájában. Például

```
> alias( nonnegative = RealRange(0,infinity),
>   nonnegbutlessthan1 = RealRange(0,Open(1)) ):
> addproperty( nonnegbutlessthan1, {nonnegative}, {0} ):
> ‘property/ParentTable‘[0];
```

$$\{nonnegbutlessthan1,\ composite\}$$

Ezzel megváltoztattuk a numerikus tulajdonságok hálóját:

```
> ‘property/ParentTable‘[nonnegbutlessthan1];
```

$$\{nonnegative\}$$

```
> ‘property/ChildTable‘[nonnegative];
```

$$\{nonnegbutlessthan1,\ \mathrm{RealRange}(\mathrm{Open}(0),\ \infty)\}$$

```
> is( 1/2, nonnegbutlessthan1 );
```

true

```
> is( Pi, nonnegbutlessthan1 );
```

false

A tulajdonságok is Maple objektumok, ezért maguk is rendelkezhetnek tulajdonságokkal. Hierarchiájuk a fejezet végén a 13.4. ábrán látható. Ezért az ilyen kérdéseknek is van értelme:

```
> is( prime, property );
```

true

Az f objektumokon értelmezett függvényhez tartozó $\overline{f}$ tulajdonságfüggvényt a `'property/f'` eljárásban implementálhatjuk. Az eljárás emlékezőtáblája a tulajdonságfüggvényre vonatkozó sok lényeges információt tárol. Vegyük például az exponenciális függvényt. A `verboseproc` interfész változót 3-ra beállítva a Maple az eljárás emlékezőtábláját is kiírja:

```
> interface( verboseproc=3 ):
> print( 'property/exp' );

proc(a)
local b, c;
option remember, 'Copyright (c) 1992 Gaston Gonnet,
Wissenschaftliches Rechnen, ETH Zurich. All rights reserved.';
    if a::RealRange then
        if op(1, a)::{0, identical(-infinity), Open(0)} and
        op(2, a)::{0, Open(0), identical(infinity)} then
            RealRange(procname(op(1, a)), procname(op(2, a)))
        else
            b := procname(op(1, a));
            c := procname(op(2, a));
            RealRange(b, c)
        fi
    elif a::EvalfableProp then PropRange(BottomProp,
        RealRange('property/Shake'(exp(a))))
    elif a::Open(EvalfableProp) then PropRange(BottomProp,
        RealRange('property/Shake'(exp(op(1, a)))))
    else ERROR(FAIL)
    fi
end
# (-infinity) = Open(0)
# (Open(0)) = Open(1)
# (real) = RealRange(Open(0), infinity)
# (complex) = complex
# (infinity) = infinity
# (0) = 1
```

A tulajdonságfüggvényt a tulajdonságokkal való számolásra használjuk, de ennél többről van szó. Például milyen következtetéseket tud levonni a Maple, ha egész argumentumra alkalmazzuk az exponenciális függvényt? A rendszer a következőképpen működik. Először is megállapítja, hogy az $\overline{\exp}$ tulajdonságfüggvény erre az esetre vonatkozóan nem tartalmaz semmi információt. Ekkor a Maple úgy dönt, hogy megvizsgálja az `integer` tulajdonság közvetlen őseit. Ámde nem talál sem a `GaussianInteger`, sem a `rational` tulajdonsághoz rendelt tulajdonságfüggvényt. A nagyszülők vizsgálatára tér át, s itt már sikerül is találnia valamit.

$$\overline{\texttt{exp}}(\texttt{complex}) = \texttt{complex}$$
$$\overline{\texttt{exp}}(\texttt{real}) = \texttt{RealRange(Open(0),infinity)}$$

Így a Maple arra a következtetésre jut, hogy

$$\begin{aligned}\overline{\texttt{exp}}(\texttt{integer}) &\subseteq \texttt{AndProp}(\overline{\texttt{exp}}(\texttt{complex}),\overline{\texttt{exp}}(\texttt{real}))\\ &= \texttt{AndProp(complex,RealRange(Open(0),infinity))}\\ &= \texttt{RealRange(Open(0),infinity)}.\end{aligned}$$

A közvetlen leszármazottak és az unokák megfigyelése alapján a Maple rájön, hogy

$$\begin{aligned}\overline{\texttt{exp}}(\texttt{integer}) &\supseteq \texttt{OrProp}(\overline{\texttt{exp}}(0),\overline{\texttt{exp}}(1))\\ &= \texttt{OrProp(1,PropRange(BottomProp,}\\ &\qquad \texttt{RealRange(2.71828182574,2.71828183118)))}\\ &\supseteq \{1\}.\end{aligned}$$

A Maple tehát biztos benne, hogy

$$\{1\} \subseteq \overline{\texttt{exp}}(\texttt{integer}) \subseteq (0,\infty)$$

Az **assume**-mal kapcsolatos függvények közül a két legfontosabb: az **assume** és az **is**. Az első függvénnyel a Maple-nek adhatunk információt az objektumokról, vagyis deklarálhatjuk az objektumok tulajdonságait. Például az

```
> assume( x>0 ):
```

parancs x-hez egy x~ lokális változót rendel, és létrehozza vagy aktualizálja a `` `property/object`[x~] `` bejegyzést. Vizsgáljuk meg ezt a táblaelemet, és nézzük meg, hogyan változik, ha egy újabb kiegészítő föltételt adunk meg:

```
> `property/object`[x];
```

$$\text{RealRange}(\text{Open}(0), \infty)$$

```
> additionally( x, integer );
> `property/object`[x];
```

$$\text{AndProp}(\mathit{integer}, \text{RealRange}(1, \infty))$$

Az objektum összes tulajdonsága tehát a `` `property/object` `` tábla megfelelő bejegyzésében tárolódik.

Objektumokra vonatkozó információkat az **is** segítségével kereshetünk vissza. Az eljárás az előző alfejezetben leírt tulajdonságalgebra szabályait alkalmazza a kifejezések tulajdonságainak ellenőrzésére. Az x-re tett előbbi föltételekből például a Maple le tudja vezetni, hogy $e^x > 1$:

```
> is( exp(x) > 1 );
```

$$\mathit{true}$$

Ez négy lépésben történik:

1. Tulajdonságok segítségével átfogalmazza a kérdést, azaz `is( exp(x~), RealRange(Open(1), infinity) )`.

2. Az objektumot kifejezi a kezdetben rendelkezésünkre álló objektumokkal, azaz `object = exp(x~)`.

3. A `'property/object'` táblából kikeresi az x~-re vonatkozó bejegyzéseket, és kiszámítja $\overline{\text{exp}}$(`'property/object'[x~]`)-et.

4. Eldönti, hogy $\overline{\text{exp}}$(`'property/object'[x~]`)-et tartalmazza-e a `RealRange(Open(1), infinity)` tulajdonság. Ebben az esetben a válasz triviálisan *true* lesz.

Az **is** eljárás így kezeli a kérdéseket:

1. Átfogalmazza a kérdést `is( object, Property )` alakúra.

2. Az objektumot kifejezi az ismert objektumokkal: `object = f( obj_1, obj_2, ..., obj_n)`.

3. Kikeresi a `'property/object'` táblából az `obj_1, obj_2, ..., obj_n` objektumokra vonatkozó bejegyzéseket, és kiszámítja `f̄(Prop_1, Prop_2, ..., Prop_n)`-t.

4. Eldönti, hogy `f̄(Prop_1,Prop_2,...,Prop_n)` benne van-e `Property`-ben.

noindent Ezt a részt az **assume** implementálásának egy furcsa következményével zárjuk. Mivel az objektumokra kirótt föltételek nem az objektumokhoz kapcsolt formában, hanem a `'property/object'` táblában tárolódnak, az objektumokat fájlba elmentve, a Maple-ből kilépve, a rendszert újraindítva és végül a fájlból az objektumokat visszatöltve minden ilyen jellegű információ elvész. A következő szekció újabb változat erre a témára:

```
> assume( x>0 ):
> 'property/object'[x]; # check the property about x
```

$$\text{RealRange}(\text{Open}(0), \infty)$$

```
> xtilde := x:  # keep a copy of x
> save x, foo:
> 'property/object'[x]; # check the property about x
```

$$\text{RealRange}(\text{Open}(0), \infty)$$

Eddig mindent rendben van. Olvassuk be a `foo` fájlt:

```
> read foo;
```

$$x := x\tilde{}$$

Még mindig jónak látszik, ámde[1]

```
> `property/object`[x];
```

$$property/object_x.$$

```
> `property/object`[xtilde];
```

$$\mathrm{RealRange}(\mathrm{Open}(0), \infty)$$

A két `x~` nem ugyanaz! A `foo` nevű fájlba a rendszer az `x := x~` értékadást mentette el. Itt `x~` olyan globális változó, amelynek neve egy tildét tartalmaz. Az az objektum, amelyre a föltételt tettük, egy `x~` nevű lokális változó, de a fájl beolvasása után rá csak az `xtilde` változóval hívatkozhatunk. Ezt úgy orvosolhatjuk, hogy nemcsak a változót, hanem a tulajdonságok táblázatát is elmentjük.

```
> save xtilde, `property/object`, foo:
> read foo:
> `property/object`[xtilde];
```

$$\mathrm{RealRange}(\mathrm{Open}(0), \infty)$$

A példa azt mutatja, hogy a Maple **assume**-mal kapcsolatos lehetőségei még lényegesen javíthatók; implementációjuk teljesebbé és robusztusabbá tehető. Mindezen furcsaságok és korlátozások ellenére fontos előrelépést jelentenek a Maple által végzett számításoknál.

13.5. Gyakorlatok

1.Számítsuk ki $\cos(n\pi)$-t és $\sin(n\frac{\pi}{2})$-t, föltéve, hogy n

- egész,
- páratlan egész,
- 3-nál kisebb páratlan pozitív egész.

2. Adjuk hozzá a $(-1, 1)$ intervallumot a numerikus tulajdonságok hierarchiájához.

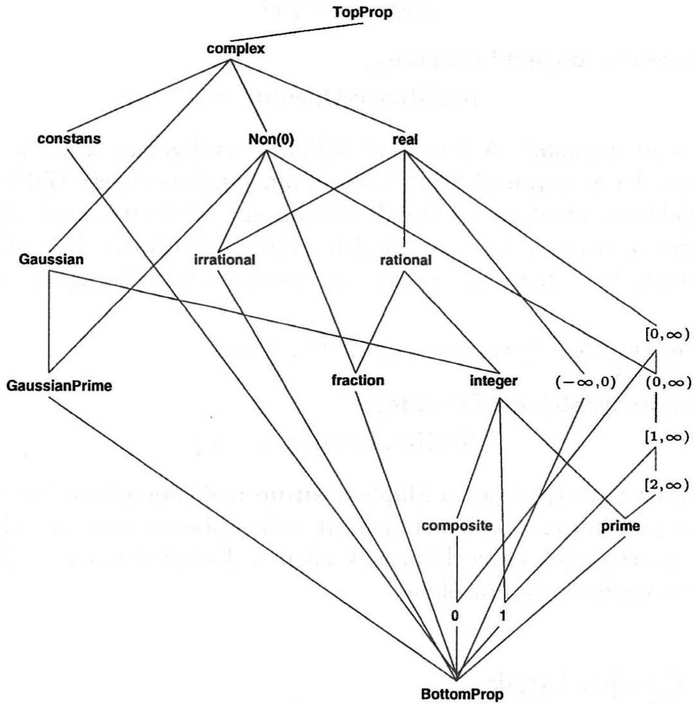
3. Egy természetes számot perfektnek nevezünk, ha eggyel nagyobb nemtriviális osztóinak összegénél (például $6 = 2 + 3 + 1$ és $28 = 2 + 4 + 7 + 14 + 1$). Vegyük hozzá a perfekt számokat a numerikus tulajdonságok hierarchiájához.

4. Az A $n \times n$-es mátrixot ferdén diagonálisnak (skew-diagonal) hívjuk, ha $A_{ij} = 0$ valahányszor $i + j \neq n + 1$. Adjuk hozzá a `skewdiagonal` tulajdonságot a Maple tudásbázisához.

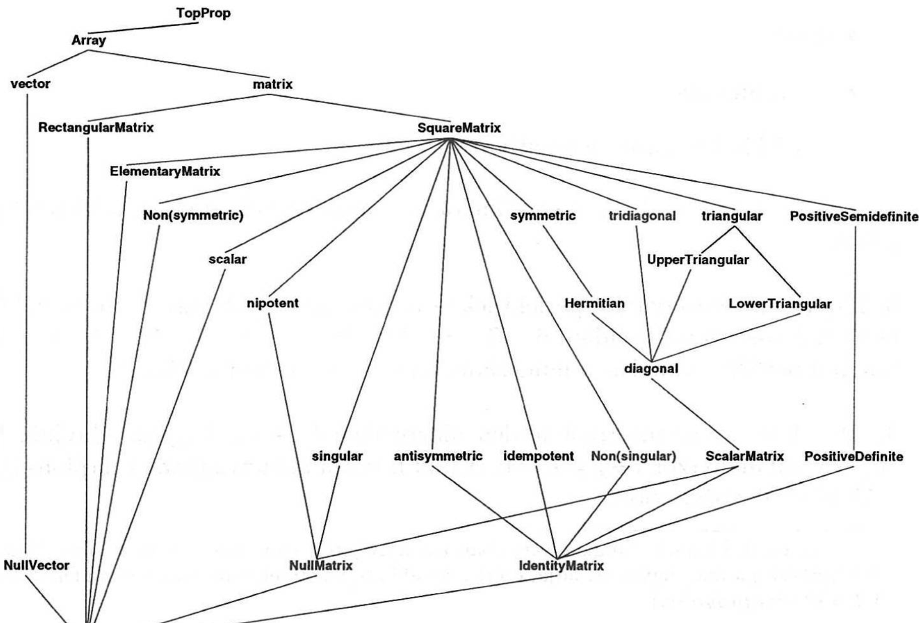
---

[1]Az általunk használt Linuxos verzióban ezt a „sajátosságot" már részben korrigálták, a két következő parancs ugyanazt adja, ezért a további fejtegetések sem igazak maradéktalanul. (*A Fordító megjegyzése.*)

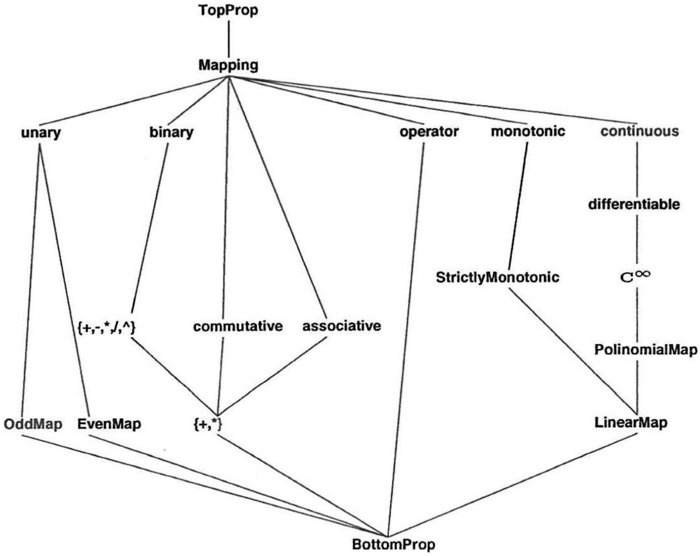
13.6. A tulajdonságok hierarchiái



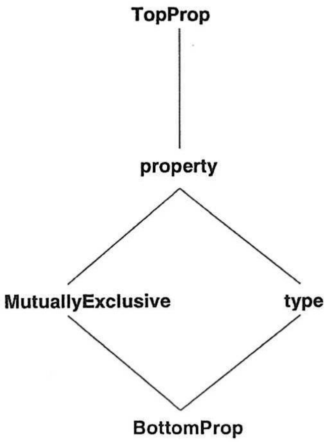
13.1. ábra: A numerikus tulajdonságok hierarchiája



13.2. ábra: A mátrixtulajdonságok hierarchiája



13.3. ábra: A függvénytulajdonságok hierarchiája



13.4. ábra: Különböző tulajdonságok hierarchiája

14. Egyszerűsítés

A 7. fejezetben tárgyaltuk a polinomokkal és a racionális kifejezésekkel végezhető műveleteket. Vizsgáltuk a normalizálást, a gyűjtést, a rendezést, a szorzattá alakítást és a kifejtést. Ezeket a műveleteket az jellemzi, hogy a kifejezések egészére hajtódnak végre. Matematikai függvényeket tartalmazó számítások során viszont gyakran olyan egyszerűsítési szabályokat szeretnénk alkalmazni, melyek ezekre a függvényekre vonatkoznak. Például trigonometrikus függvényeket használó számítások során gyakran föl szeretnénk használni a $\sin^2 + \cos^2 = 1$ egyenlőséget. Ebben a fejezetben leírjuk, hogyan hajtható végre matematikai függvényeket tartalmazó kifejezések egyszerűsítése, mit mondhatunk a Maple által fölajánlott egyszerűsítések érvényességéről, hogyan kontrollálhatók az egyszerűsítések, hogyan definiálhatunk saját egyszerűsítési rutinokat, vagy bírálhatjuk fölül a Maple meglévő rutinjait.

Az automatikus egyszerűsítést bemutató rövid bevezető után az **expand**, a **combine**, a **simplify** és a **convert** eljárásokat ismertetjük. A Maple specialitásai közé tartozik, hogy maga a rendszer határozza meg, hogy milyen matematikai függvények vesznek részt a számításokban, és milyen ezekre vonatkozó transzformációs szabályok állnak rendelkezésére. Ez újabb példát szolgáltat a Maple rendszer ***hibrid algoritmikus struktúrájára.***

Megvizsgáljuk minden egyes egyszerűsítési eljárás hatását a trigonometrikus függvényekre, az exponenciális és logaritmus függvényekre, a hatványokra, a gyökös kifejezésekre és más függvényekre. Három külön alfejezetet szentelünk a trigonometrikus függvények egyszerűsítésének, a kifejezések mellékföltételeket is figyelembe vevő egyszerűsítésének és az egyszerűsítési folyamat kontrollálásának. Azt is megmutatjuk, hogyan definiálhatunk saját egyszerűsítési rutinokat vagy bírálhatjuk fölül a Maple által végzett egyszerűsítéseket. A fejezet végén a lehetséges egyszerűsítéseket áttekintő táblázatot közlünk (lásd 14.1. táblázat).

14.1. Automatikus egyszerűsítés

Vessünk először egy pillantást a Maple által végrehajtott automatikus egyszerűsítésekre:

```
> arcsin(1/2), Zeta(2), GAMMA(1/2), Psi(1/2);
```

$$\frac{1}{6}\pi,\ \frac{1}{6}\pi^2,\ \sqrt{\pi},\ -\gamma - 2\ln(2)$$

```
> min( 3, Pi, cos(1)*3 );
```

$$3\cos(1)$$

```
> tan( arctan(x) ), arctan( tan(x) );
```

$$x,\ \arctan(\tan(x))$$

```
> sqrt(Pi^2*x^2);
```

$$\pi\sqrt{x^2}$$

```
> 'abs( abs(x) )' = abs( abs(x) );
```

$$||x|| = |x|$$

```
> 'abs(-Pi*x)' = abs(-Pi*x);
```

$$|-\pi\, x| = \pi\, |x|$$

```
> 'sin(-x)' = sin(-x), 'cos(Pi/2+x)' = cos(Pi/2+x);
```

$$\sin(-x) = -\sin(x),\ \cos(\frac{1}{2}\pi + x) = -\sin(x)$$

```
> 'exp(3*ln(x))' = exp(3*ln(x));
```

$$e^{(3\ln(x))} = x^3$$

```
> ln(exp(x)), exp(ln(x));
```

$$\ln(e^x),\ x$$

```
> assume( x, 'real' );
> ln( exp(x) );
```

$$x\tilde{}$$

```
> signum( exp(1)+exp(x));
```

$$1$$

Az 1.3. alfejezetben már figyelmeztettük az Olvasót, hogy nem minden automatikus egyszerűsítési szabály érvényes általánosan. Néha csak az alapesetekben működnek helyesen. Például a $0 \cdot x \longrightarrow 0$ és az $(x - x) \longrightarrow 0$ egyszerűsítések (hatékonysági megfontolásból) föltétlenül kívánatosak, de ezek sem adnak helyes eredményt, ha x értéke definiálatlan vagy végtelen. Az $(x - x) \longrightarrow 0$ egyszerűsítésnek a következő mellékhatása van:

```
> limit( exp(1/x), x=0 ) - limit( exp(-1/x), x=0 );
```

$$0$$

A Maple által minden k-ra végrehajtott $0^k \longrightarrow 0$ automatikus egyszerűsítés pedig az alábbi mellékhatást eredményezi:

```
> sum( a[k]*x^k, k = 0..n );
```

$$\sum_{k=0}^{n} a_k\, x^k$$

```
> eval(subs( x=0, " ));
```

$$0$$

Az $\exp(\ln(x)) \longrightarrow x$ automatikus egyszerűsítés helytelen az $x = 0$ esetben, de ettől még a Maple habozás nélkül alkalmazza.

A fenti példák jól mutatják egyrészt a használhatóság és a hatékonyság, másrészt a matematikai egzaktság közti kapcsolatot. Tisztában kell lennünk azzal, hogy a Maple a legtöbb vagy talán az összes számítógépes algebrai rendszerhez hasonlóan olyan automatikus egyszerűsítési szabályokat hajt végre, amelyek nem száz százalékosan biztonságosak.

Sok rendszerben a problémák újabb forrását jelenti a főértékükkel reprezentált többértékű függvények automatikus egyszerűsítése. A fejlesztők komoly erőfeszítéseket tettek a komplex többértékű függvények korrekt kezelésére a Maple V Release 4-ben. Csak a bizonyíthatóan helyes automatikus egyszerűsítési szabályok hajtódnak végre (a korábban említett $\exp(\ln(x)) \longrightarrow x$-et kivéve). Például az

```
> sqrt(Pi^2*x^2);
```

$$\pi \sqrt{x^2}$$

kifejezést nem egyszerűsíti tovább a rendszer.

```
> sqrt( Pi^2*x^2, 'symbolic' );
```

$$\pi\, x$$

Ehhez vagy további föltételeket kell tennünk, vagy a fönti példához hasonlóan a `symbolic` kulcsszót kell használnunk, hogy valamely nem verifikálható egyszerűsítésre rávegyük a rendszert.

De óvatosaknak kell lennünk, és tudomásul kell venni, hogy a többértékű függvények egyszerűsítése még mindig nem száz százalékig biztonságos. A fejezetből ki fog derülni, hogy a Maple megengedi olyan egyszerűsítések használatát, amelyek csak az alapesetekben és nem minden elképzelhető esetben érvényesek. Egy ijesztő példa:

```
> ln( exp(x) ); # no automatic simplification
```

$$\ln(e^x)$$

```
> simplify( ", ln ); # no harm done
```

$$\ln(e^x)$$

```
> simplify( ", exp ); # no harm done
```

$$\ln(e^x)$$

```
> simplify( ", {ln,exp} ); # possibly incorrect
```

$$x$$

Az utóbbi helyettesítés csak akkor korrekt, ha az x komplex szám argumentuma $-\pi$ és π között van (a határokat is beleértve). Azt mondhatnánk, hogy ha elég erőteljesen noszogatjuk, a Maple megadja a kívánt eredményeket, függetlenül attól, hogy ezek matematikailag helyesek-e. De az ilyen egyszerűsítések gyakran hibás számítási eredményekre vezetnek. Ha az Olvasó az eddigiek alapján még nem lenne meggyőződve arról, hogy a programot kellő óvatossággal kell használnia, akkor vessen egy pillantást az [57, 111, 175] publikációkra, ahol hasonlóan meglepő eredményeket találhat.

14.2. Az **expand** eljárás

Az expand eljárásról általánosságban azt mondhatjuk, hogy nevéhez hűen dolgozik, nevezetesen kifejezéseket fejt ki. Matematikai függvények estében ez gyakran az addíciós szabályok alkalmazását jelenti.

- **Trigonometrikus és hiperbolikus függvények**

```
> cos(2*x): " = expand(");
```

$$\cos(2\,x) = 2\cos(x)^2 - 1$$

```
> cos(5*x): " = expand(");
```

$$\cos(5\,x) = 16\cos(x)^5 - 20\cos(x)^3 + 5\cos(x)$$

```
> cosh(5*x): " = expand(");
```

$$\cosh(5\,x) = 16\cosh(x)^5 - 20\cosh(x)^3 + 5\cosh(x)$$

```
> tan(2*x): " = expand(");
```

$$\tan(2\,x) = 2\,\frac{\tan(x)}{1-\tan(x)^2}$$

```
> tanh(2*x): " = expand(");
```

$$\tanh(2\,x) = 2\,\frac{\sinh(x)\cosh(x)}{2\cosh(x)^2 - 1}$$

```
> cos(x+y): " = expand(");
```

$$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$$

```
> cos(x+2*y): " = expand(");
```

$$\cos(x+2\,y) = 2\cos(x)\cos(y)^2 - \cos(x) - 2\sin(x)\sin(y)\cos(y)$$

```
> cos(x*(y+z)): " = expand(");
```

$$\cos(x\,(y+z)) = \cos(x\,y)\cos(x\,z) - \sin(x\,y)\sin(x\,z)$$

Az előző példa két szembetűnő aspektusa:

(1) Az **expand** eljárás kifejti az összegek vagy többszörös argumentumok trigonometrikus és hiperbolikus függvényeit. A fönti példák is azt illusztrálják, hogy a kifejtés olyan mélységig történik, ameddig csak lehetséges. A $\cos 5x \longrightarrow \cos x \cos 4x - \sin x \sin 4x$-hez hasonló részleges kifejtést már nem olyan könnyű végrehajtani az **expand**-dal. Ez csak az alábbiakhoz hasonló trükkökkel érhető el:

```
> cos(5*x);
```

$$\cos(5\,x)$$

```
> subs( 5*x = x + y, " );
```

$$\cos(x + y)$$

```
> expand(");
```

$$\cos(x)\cos(y) - \sin(x)\sin(y)$$

```
> subs( y = 4*x, " );
```

$$\cos(4\,x)\cos(x) - \sin(x)\sin(4\,x)$$

Vagy pedig az **expand/cos** alább látható forráskódja alapján át kell írni a kifejtést végző eljárást; ki kell hagyni belőle önmaga rekurzív hívását:

```
> interface( verboseproc=2 ): # make Maple more verbose
> print( readlib( `expand/cos` ) ); # print source code

proc(x)
local n, y;
option `Copyright (c) 1991 by the University of Waterloo`
  y := expand(x);
  if type(y, `+`) then
    n := op(1, y); y := y - n;
    expand(cos(n)*cos(y) - sin(n)*sin(y))
  elif type(y, `*`) then
    n := op(1, y);
    if type(n, numeric) and n < 0 then expand(cos(-y))
    elif type(n, integer) and 0 < n and n < 100 then
      y := y/n;
      expand(2*cos((n - 1)*y)*cos(y) - cos((n - 2)*y))
    else cos(y)
    fi
  else cos(y)
  fi
end

> expand( cos(5*x) ); # an example
```

$$16\cos(x)^5 - 20\cos(x)^3 + 5\cos(x)$$

Az új eljárás kódja például így írható meg:

```
> `expand/cos` := proc(x)
> local n,y;
>   y := expand(x);
>   if type(y,`+`) then
>     n := op(1,y);
>     y := y-n;
>     cos(n)*cos(y)-sin(n)*sin(y)
>   elif type(y,`*`) then
>     n := op(1,y);
>     if type(n,numeric) and n < 0 then expand(cos(-y))
>     elif type(n,integer) and 0 < n and n < 100 then
>       y := y/n;
>       2*cos((n-1)*y)*cos(y)-cos((n-2)*y)
>     else cos(y)
>     fi
>   else cos(y)
>   fi
> end:
```

Az új változat nem vezetne azonnali eredményre, mivel az **expand** még emlékezik a $\cos 5x$ korábbi $16\cos^5 x - 20\cos^3 x + 5\cos x$ kifejtésére. A **forget**-et a `reinitialize=false` opcióval kell meghívnunk, hogy az emlékezőtábla törlődjön, de ne olvassa be újra a rendszer a rutin eredeti változatát:

```
> readlib(forget)( expand, reinitialize = false ):
> expand( cos(5*x) );
```

$$2\cos(4\,x)\cos(x) - \cos(3\,x)$$

Egyébként az **expand** `remember` opciója magyarázza meg azt is, hogy a következő rekurziós lépésben

```
> expand(");
```

$$16\cos(x)^5 - 16\cos(x)^3 + 3\cos(x) - 2\cos(2\,x)\cos(x)$$

miért nem fordulnak már elő $\cos(3x)$-es tagok. A 14.9. alfejezetben fogjuk megvizsgálni, hogyan változtathatók meg a meglévő Maple rutinok.

(2) A tangenst (és a kotangenst) tangenseket tartalmazó racionális kifejezésekké fejti ki a rendszer, viszont a megfelelő hiperbolikus függvényeket a sinh-t és a cosh-t tartalmazó kifejezésekkel írja föl.

```
> tan(x+y):  " = expand(");
```

$$\tan(x+y) = \frac{\tan(x)+\tan(y)}{1-\tan(x)\tan(y)}$$

```
> tanh(x+y):  " = expand(");
```

$$\tanh(x+y) = \frac{\sinh(x)\cosh(y)+\cosh(x)\sinh(y)}{\cosh(x)\cosh(y)+\sinh(x)\sinh(y)}$$

A 14.9. alfejezetben megnézzük, hogyan definiálható át úgy a hiperbolikus tangens kifejtési rutinja, hogy az eredmény konzisztens legyen a megfelelő trigonometrikus függvényre kapottal.

- **exp, ln**

```
> exp(x+y): " = expand(");
```

$$e^{(x+y)} = e^x\, e^y$$

```
> ln(x*y): " = expand(");
```

$$\ln(x\,y) = \ln(x) + \ln(y)$$

```
> ln(x^y): " = expand(");
```

$$\ln(x^y) = y\ln(x)$$

```
> ln(x/y): " = expand(");
```

$$\ln(\frac{x}{y}) = \ln(x) - \ln(y)$$

```
> exp(x*(y+z)): " = expand(");
```

$$e^{(x\,(y+z))} = e^{(x\,y)}\, e^{(x\,z)}$$

Annak felelőssége, hogy egy transzformációs szabály érvényes-e vagy sem, a miénk. Például az $\ln(xy) \longrightarrow \ln x + \ln y$ transzformációs szabály általában nem igaz (nézzük meg, mi történik, ha az $x = y = -1$ helyettesítést alkalmazzuk), a rendszer kérésünkre mégis bármikor hajlandó alkalmazni.

- **Gyökök és hatványok**

```
> x^(y+z): " = expand(");
```

$$x^{(y+z)} = x^y\, x^z$$

```
> (x*y)^z: " = expand(");
```

$$(x\,y)^z = x^z\, y^z$$

```
> (-x)^y: " = expand(");
```

$$(-x)^y = (-1)^y\, x^y$$

```
> (x/y)^z: " = expand(");
```

$$(\frac{x}{y})^z = x^z\,(\frac{1}{y})^z$$

```
> x^(y/3): " = expand(");
```

$$x^{(1/3\,y)} = (x^y)^{1/3}$$

```
> (x^(1/2))^(y/2): " = expand(");
```

$$(\sqrt{x})^{(1/2\,y)} = x^{(1/4\,y)}$$

Megismételjük, hogy az **expand** által végrehajtott transzformácók közül nem mindegyik száz százalékig biztonságos. Például az $(x\,y)^z \longrightarrow x^z\,y^z$ transzformácó sem érvényes általában, erről meggyőződhetünk az $x = y = -1$, $z = 1/2$ helyettesítéssel.

```
> (x*y)^z:  " = expand(");
```

$$(x\,y)^z = x^z\,y^z$$

```
> subs( { x=-1, y=-1, z=1/2}, " );
```

$$1 = -1$$

A fölhasználónak kell meggyőződnie arról, hogy a transzformácó egy adott esetben érvényes-e. Még inkább résen kell lennünk, ha valamely egyszerűsítéseket is tartalmazó szimbolikus számítás után helyettesítünk be konkrét értékeket a végeredménybe. Nem biztos, hogy az ekkor beírt értékekre is korrekt volt a számolás közben korábban alkalmazott összes átalakítás. Efféle félreértések elég gyakran előfordulnak; a jelenség neve *specializációs probléma.*

• További kifejtések

Az **expand** eljárás további egyszerűsítéseket is végez: szorzattá alakított természetes számok, a faktoriális és a binomiális együtthatók, valamint a Gamma függvény kifejtését, végül mátrixszorzatok és mátrixhatványok egyszerűsítését:

```
> (n+1)!:  " = expand(");
```

$$(n+1)! = n!\,(n+1)$$

```
> binomial(n+1,k+1):  " = expand(");
```

$$\mathrm{binomial}(n+1,\,k+1) = \frac{(n+1)\,\mathrm{binomial}(n,\,k)}{k+1}$$

```
> binomial(n-1,k-1) + binomial(n-1,k);
```

$$\mathrm{binomial}(n-1,\,k-1) + \mathrm{binomial}(n-1,\,k)$$

```
> expand(");
```

$$\frac{k\,\mathrm{binomial}(n,\,k)}{n} + \frac{(n-k)\,\mathrm{binomial}(n,\,k)}{n}$$

```
> normal(");
```

$$\mathrm{binomial}(n,\,k)$$

```
> ifactor(123456789):  " = expand(");
```

$$(\,3)^2\,(\,3803)\,(\,3607) = 123456789$$

```
> BesselJ(5,t):  " = expand(");
```

$$\mathrm{BesselJ}(5,\,t) = 384\,\frac{\mathrm{BesselJ}(1,\,t)}{t^4} - 192\,\frac{\mathrm{BesselJ}(0,\,t)}{t^3} - 72\,\frac{\mathrm{BesselJ}(1,\,t)}{t^2} + 12\,\frac{\mathrm{BesselJ}(0,\,t)}{t} + \mathrm{BesselJ}(1,\,t)$$

```
> collect( ", BesselJ ); # group terms
```

$$\mathrm{BesselJ}(5,\,t) = \left(\frac{384}{t^4} - \frac{72}{t^2} + 1\right)\mathrm{BesselJ}(1,\,t) + \left(-\frac{192}{t^3} + \frac{12}{t}\right)\mathrm{BesselJ}(0,\,t)$$

```
> Zeta(50): " = expand(");
```

$$\zeta(50) = \frac{39604576419286371856998202}{285258771457546764463363635252374414183254365234375}\pi^{50}$$

```
> dilog(1/x): " = expand(");
```

$$\mathrm{dilog}\left(\frac{1}{x}\right) = -\mathrm{dilog}(x) - \frac{1}{2}\ln(x)^2$$

```
> M := array(1..4,1..4):
> det(M^3): " = expand(");
```

$$\det(M^3) = \det(M)^3$$

```
> det(3*M): " = expand(");
```

$$\det(3\,M) = 81\det(M)$$

Azt kell még elmondanunk, hogyan előzhető meg valamely kifejezésben szereplő adott nemracionális függvény kifejtése. Ez azért fontos, mert egy szimbolikus kifejezésben több matematikai függvény is előfordulhat, és nem biztos, hogy midegyiket ki szeretnénk fejteni. Például tegyük föl, hogy a $\sin(x+y)+\exp(x+y)$ kifejezésben csak az exponenciális függvényt akarjuk kifejteni. Ha csak egyszerűen alkalmazzuk az **expand**-ot, a teljesen kifejtett alakot kapjuk. Csak akkor marad változatlan a trigonometrikus függvény, ha a hívás extra argumentumaként megadjuk a `sin` kulcsszót:

```
> expression := sin(x+y) + exp(x+y);
```

$$\mathit{expression} := \sin(x+y) + e^{(x+y)}$$

```
> expand( expression );
```

$$\sin(x)\cos(y) + \cos(x)\sin(y) + e^x\,e^y$$

```
> expand( expression, sin );
```

$$\sin(x+y) + e^x\,e^y$$

Ez megegyezik a 7.1. alfejezetben leírt mechanizmussal: az **expand** extra argumentumai azt adják meg, hogy mit kell érintetlenül hagyni:

```
> expand( sin( x^sin(y+z) + w ), x^sin(y+z) );
```

$$\sin(x^{\sin(y+z)})\cos(w) + \cos(x^{\sin(y+z)})\sin(w)$$

```
> expand( ( cos(2*x) + sin(2*y) )^2, cos, sin );
```

$$\cos(2\,x)^2 + 2\cos(2\,x)\sin(2\,y) + \sin(2\,y)^2$$

Amennyiben az *összes* nemracionális függvény kifejtését el akarjuk kerülni, a **frontend** Maple eljárást használhatjuk:

```
> frontend( expand, [expression^2] );
```

$$\sin(x+y)^2 + 2\sin(x+y)\,e^{(x+y)} + (e^{(x+y)})^2$$

Ha előre tudjuk, hogy egy ideig nem lesz szükségünk bizonyos nemracionális függvények kifejtésére, a Maple-t az **expandoff** eljárással tájékoztathatjuk erről:

```
> expand( expandoff() ): # enable library function
> expandoff( sin ): # turn off expansion of sin
> expression := sin(p+q) + exp(p+q);
```

$$expression := \sin(p+q) + e^{(p+q)}$$

```
> expand( expression );
```

$$\sin(p+q) + e^p\,e^q$$

Talán (még mindig) csodálkozik az Olvasó azon, hogy miért írtuk föl újra az előző kifejezést, melyben eredetileg x és y szerepelt, a p és a q új ismeretlenekkel. Ha nem ezt tettük volna, akkor a Maple emlékezett volna, hogy az **expand**(`expression`) parancsot már beírtuk egyszer, következésképpen az **expand** eljárás emlékezőtáblájából elővette volna az előző parancs eredményét, ahelyett, hogy újra kiszámolta volna. Győződjünk meg erről az **expandoff**-fal ellentétes hatású **expandon** megadásával:

```
> expandon( sin ): # turn on expansion of sin
> expand( expression );
```

$$\sin(p+q) + e^p\,e^q$$

Mint korábban láttuk, a **forget** könyvtári függvényt is alkalmazhattuk volna az **expand** emlékezőtáblájának törlésére.

A 14.8. alfejezetben részletesebben megvizsgáljuk, hogyan kontrollálhatjuk az egyszerűsítéseket.

14.3. A combine eljárás

Ha egy számítógépes algebrai rendszer lehetőséget kínál kifejezések kifejtésére, akkor elvárhatjuk olyan eljárás létezését is, amely ennek az ellenkezőjét végzi, azaz egyesít kifejezéseket. Ne feledjük, hogy a Maple gyakran a fölhasználóra bízza a végrehajtott átalakítások helyességének ellenőrzését.

```
> sqrt(x) * sqrt(y): " = combine(");
```

$$\sqrt{x}\,\sqrt{y} = \sqrt{x\,y}$$

Ez az egyenlőség sem igaz minden x és y értékre.

```
> subs( {x=-1,y=-1}, " );
```

$$-1 = 1$$

Másik példánk a divergens összegekre alkalmazott **combine**:

```
> Sum( 1/(2*k), k=1..infinity ) -
> Sum( 1/(2*k-1), k=1..infinity );
```

$$\left(\sum_{k=1}^{\infty}\left(\frac{1}{2}\,\frac{1}{k}\right)\right)-\left(\sum_{k=1}^{\infty}\frac{1}{2\,k-1}\right)$$

```
> value("); # correct error message
Error, (in value) invalid cancellation of infinity
> combine("); # combine sums without checking conditions
```

$$\sum_{k=1}^{\infty}\left(\frac{1}{2}\,\frac{1}{k}-\frac{1}{2\,k-1}\right)$$

```
> value(");
```

$$-\ln(2)$$

- **Trigonometrikus és hiperbolikus függvények**

A **combine** a szinuszok és koszinuszok polinomjait a

$$\begin{aligned}
\sin x \sin y &\longrightarrow \frac{1}{2}\cos(x-y)-\frac{1}{2}\cos(x+y)\\
\sin x \cos y &\longrightarrow \frac{1}{2}\sin(x-y)+\frac{1}{2}\sin(x+y)\\
\cos x \cos y &\longrightarrow \frac{1}{2}\cos(x-y)+\frac{1}{2}\cos(x+y)
\end{aligned}$$

szabályok ismételt alkalmazásával trigonometrikus polinomokká alakítja.

```
> 2*sin(x)*cos(x): " = combine(");
```

$$2\sin(x)\cos(x)=\sin(2\,x)$$

```
> sin(x)^3: " = combine(");
```

$$\sin(x)^3=-\frac{1}{4}\sin(3\,x)+\frac{3}{4}\sin(x)$$

Hasonló szabályok érvényesek a hiperbolikus függvényekre is.

$$\begin{aligned}
\sinh x \sinh y &\longrightarrow \frac{1}{2}\cosh(x-y)-\frac{1}{2}\cosh(x+y)\\
\sinh x \cosh y &\longrightarrow \frac{1}{2}\sinh(x-y)+\frac{1}{2}\sinh(x+y)\\
\cosh x \cosh y &\longrightarrow \frac{1}{2}\cosh(x-y)+\frac{1}{2}\cosh(x+y)
\end{aligned}$$

```
> cosh(x)^5: " = combine(");
```

$$\cosh(x)^5=\frac{1}{16}\cosh(5\,x)+\frac{5}{16}\cosh(3\,x)+\frac{5}{8}\cosh(x)$$

Ha kizárólag a trigonometrikus függvényekre akarjuk alkalmazni a **combine**-t, a `trig` opciót kell megadni. Hasonlítsuk össze az alábbi eredményeket:

```
> sqrt(cos(x)^2) * sqrt(sin(x)^2);
```

$$\sqrt{\cos(x)^2}\,\sqrt{\sin(x)^2}$$

```
> " = combine(");
```

$$\sqrt{\cos(x)^2}\,\sqrt{\sin(x)^2} = \sqrt{\cos(x)^2 - \cos(x)^4}$$

```
> "" = combine( "", 'trig' );
```

$$\sqrt{\cos(x)^2}\,\sqrt{\sin(x)^2} = \frac{1}{4}\sqrt{2\cos(2\,x)+2}\,\sqrt{2-2\cos(2\,x)}$$

Eléggé furcsa módon a hiperbolikus függvények esetében is a `trig` kulcsszót kell használni a természetesnek tűnő `trigh` helyett:

```
> sqrt( cosh(x)^2-1) * sqrt(cosh(x)^2+1);
```

$$\sqrt{\cosh(x)^2-1}\,\sqrt{\cosh(x)^2+1}$$

```
> " = combine( ", 'trig' );
```

$$\sqrt{\cosh(x)^2-1}\,\sqrt{\cosh(x)^2+1} = \frac{1}{4}\sqrt{2\cosh(2\,x)-2}\,\sqrt{2\cosh(2\,x)+6}$$

- **exp, ln**

A **combine** az exponenciális és a logaritmus függvényeket tartalmazó kifejezéseket a következő szabályoknak megfelelően alakítja át:

$\exp x \exp y$	$\longrightarrow$	$\exp(x+y)$
$\exp(x+n\ln y)$	$\longrightarrow$	$y^n \exp(x)$ minden $n \in \mathbb{Z}$-re
$(\exp x)^n$	$\longrightarrow$	$\exp(nx)$ minden $n \in \mathbb{Z}$-re
$y \ln x$	$\longrightarrow$	$\ln(x^y)$, ha $y \in \mathbb{Q}$ és $y \arg(x) = \arg(x^y)$
$\ln x + \ln y$	$\longrightarrow$	$\ln(x\,y)$, ha $\arg(x) + \arg(y) = \arg(x\,y)$.

A logaritmusok esetében a Maple csak a bizonyíthatóan helyes kombinációkat hajtja végre:

```
> exp(x)* exp(y)^(2):  " = combine(");
```

$$e^x\,(e^y)^2 = e^{(x+2\,y)}$$

```
> x*ln(2) + 3*ln(x)+4*ln(5):  " = combine(");
```

$$x\ln(2) + 3\ln(x) + 4\ln(5) = x\ln(2) + 3\ln(x) + \ln(625)$$

```
> ln(x) + ln(2) + ln(y) - ln(3):  " = combine(");
```

$$\ln(x) + \ln(2) + \ln(y) - \ln(3) = \ln(2\,x) + \ln(\frac{1}{3}\,y)$$

```
> exp( x - 2*ln(y) ):  " = combine(");
```

$$e^{(x-2\ln(y))} = \frac{e^x}{y^2}$$

Ha tényleg azt szeretnénk, hogy a Maple az $\ln x + \ln y \longrightarrow \ln(x\,y)$ kombinációt alkalmazza, ezt a **combine** hívásában megadott `symbolic` kulcsszóval kényszeríthetjük ki:

```
> expression := ln(x) + 1/2*ln(y) - ln(z);
```

$$\mathit{expression} := \ln(x) + \frac{1}{2}\ln(y) - \ln(z)$$

```
> combine( expression, ln ); # by default, nothing done
```

$$\ln(x) + \frac{1}{2}\ln(y) - \ln(z)$$

```
> expression: " = combine( ", ln, 'symbolic' );
```

$$\ln(x)+\frac{1}{2}\ln(y)-\ln(z)=\ln(\frac{x\sqrt{y}}{z})$$

Valójában ennél nagyobb befolyásunk is van az egyszerűsítések végrehajtására. Például a **combine** alkalmazását egész együtthatós logaritmusos tagokra is korlátozhatjuk:

```
> expression: " = combine( ", ln, integer, 'symbolic' );
```

$$\ln(x)+\frac{1}{2}\ln(y)-\ln(z)=\frac{1}{2}\ln(y)+\ln(\frac{x}{z})$$

```
> expression: " = combine( ", ln, 'positive', 'symbolic' );
```

$$\ln(x)+\frac{1}{2}\ln(y)-\ln(z)=-\ln(z)+\ln(x\sqrt{y})$$

Az általános esetben jól alkalmazható az `anything` típus:

```
> expression := m*ln(x) + n*ln(y) + 3*ln(z);
```

$$expression := m\ln(x)+n\ln(y)+3\ln(z)$$

```
> expression: " = combine( ", ln, 'symbolic' );
```

$$m\ln(x)+n\ln(y)+3\ln(z)=m\ln(x)+n\ln(y)+\ln(z^3)$$

```
> expression: " = combine( ", ln, string, 'symbolic' );
```

$$m\ln(x)+n\ln(y)+3\ln(z)=3\ln(z)+\ln(x^m\,y^n)$$

```
> expression: " = combine( ", ln, anything,
> 'symbolic' );
```

$$m\ln(x)+n\ln(y)+3\ln(z)=\ln(x^m\,y^n\,z^3)$$

Ha n típusától függetlenül el szeretnénk végeztetni a **combine**-nal az

$$n\ln y\longrightarrow\ln(y^n)$$

és az

$$\exp(x+n\ln y)\longrightarrow y^n\exp(x)$$

transzformációkat, erre a következő lehetőségeink vannak:

```
> y*ln(x): combine( ", ln, anything, 'symbolic' );
```

$$\ln(x^y)$$

```
> exp( x+n*ln(y) ):
> " = map( combine, ", ln, anything, 'symbolic' );
```

$$e^{(x+n\ln(y))}=e^{(x+\ln(y^n))}$$

```
> combine(");
```

$$e^{(x+n\ln(y))}=y^n\,e^x$$

- **Gyökök és hatványok**

A **combine** hatványokra vonatkozó két legfontosabb szabálya:

$$x^y\,x^z \longrightarrow x^{y+z}$$
$$(x^y)^z \longrightarrow x^{yz}$$

A **combine** Maple eljárásnak nem erőssége a kifejezésekben előforduló hatványok fölismerése. Például a

```
> combine( x^y * x^z );
```

$$x^y\,x^z$$

hatástalan, de

```
> combine( x^y * x^z, 'power' );
```

$$x^{(y+z)}$$

már megoldja a feladatot. A `symbolic` kulcsszóval olyan transzformációkat is kikényszeríthetünk, amelyek nem száz százalékig érvényesek:

```
> combine( (x^y)^z, 'power' );
```

$$(x^y)^z$$

```
> " = combine( ", 'power', 'symbolic' );
```

$$(x^y)^z = x^{(y\,z)}$$

Ebbe a csoportba tartozik a racionális kitevős hatványokra vonatkozó `radical` opció is. Ez főleg az $x^{(m/d)}y^{(n/d)} \longrightarrow (x^m y^n)^{(1/d)}$ transzformációval kapcsolatos, ahol x és y pozitív, m, n és d pedig olyan egészek, amelyekre $|m| < d$, $|n| < d$, valamint $d > 1$ teljesül. A kulcsszó elhagyásakor néha a **combine** túl messze megy a gyökökre vonatkozó egyszerűsítéseknél:

```
> x^(1/4) * y^(1/4);
```

$$x^{1/4}\,y^{1/4}$$

```
> combine( ", radical ); # no simplification
```

$$x^{1/4}\,y^{1/4}$$

```
> combine("); # too much simplification
```

$$(x\,y)^{1/4}$$

Nézzünk meg néhány további példát a gyökök és a hatványok egyszerűsítésére:

```
> (1/2)^m * (1/2)^n:  " = combine( ", 'power' );
```

$$(\frac{1}{2})^m\,(\frac{1}{2})^n = 2^{(-m-n)}$$

```
> combine( (x^y)^z, 'power' );
```

$$(x^y)^z$$

```
> " = combine( ", 'power', 'symbolic' );
```

$$(x^y)^z = x^{(y\,z)}$$

```
> x^y / x^(2/3):  " = combine( ", 'power' );
```

$$\frac{x^y}{x^{2/3}} = x^{(y-2/3)}$$

```
> 2^(1/3) * (x+1)^(1/3):  " = combine( ", radical );
```

$$2^{1/3}(x+1)^{1/3} = (2\,x+2)^{1/3}$$

- **Egyéb kombinációk**

A **combine** alkalmazható még a négyzetgyök, az `arctan` és a `polylog` függvényt tartalmazó kifejezésekre, valamint formulák átrendezésével a negatív előjelek kiküszöbölésére:

```
> sqrt(x) * sqrt(y):  " = combine(");
```

$$\sqrt{x}\sqrt{y} = \sqrt{x\,y}$$

```
> arctan(x) + arctan(1/x):  " = combine(");
```

$$\arctan(x) + \arctan(\frac{1}{x}) = \frac{1}{2}\operatorname{signum}(x)\,\pi$$

```
> arctan(1/2) + arctan(1/3):  " = combine(");
```

$$\arctan(\frac{1}{2}) + \arctan(\frac{1}{3}) = \frac{1}{4}\,\pi$$

```
> combine( arctan(x) + arctan(y) );
```

$$\arctan(x) + \arctan(y)$$

```
> " = combine( ", arctan, 'symbolic' );
```

$$\arctan(x) + \arctan(y) = \arctan(\frac{x+y}{1-x\,y})$$

```
> polylog(2,z) + polylog(2,1-z):  " = combine(");
```

$$\operatorname{polylog}(2,\,z) + \operatorname{polylog}(2,\,1-z) = \frac{1}{6}\,\pi^2 - \ln(z)\ln(1-z)$$

```
> -(-x+1)/x:  " = combine(");
```

$$-\frac{-x+1}{x} = \frac{x-1}{x}$$

14.4. A **simplify** eljárás

A **simplify** a Maple általános célú egyszerűsítési rutinja.

- **Trigonometrikus és hiperbolikus függvények**

A **simplify** eljárás a következő szabályoknak megfelelően normalizálja az olyan racionális kifejezéseket, melyekben trigonometrikus és hiperbolikus függvények fordulnak elő:

$$\begin{aligned}
\sin^2 x &\longrightarrow 1-\cos^2 x\\
\sinh^2 x &\longrightarrow \cosh^2 x - 1\\
\tan x &\longrightarrow \frac{\sin x}{\cos x}\\
\tanh x &\longrightarrow \frac{\sinh x}{\cosh x}
\end{aligned}$$

Pontosabban a sin és a sinh függvények egynél nagyobb kitevős hatványait addig egyszerűsíti a fönti szabályok szerint, ameddig csak lehetséges. Az utolsó két szabályt csak akkor alkalmazza, ha más trigonometrikus függvények is előfordulnak:

```
> cosh(x)^2 - sinh(x)^2:  " = simplify(");
```

$$\cosh(x)^2 - \sinh(x)^2 = 1$$

```
> sinh(x)^3:  " = simplify(");
```

$$\sinh(x)^3 = -\sinh(x) + \sinh(x)\cosh(x)^2$$

```
> 2*sin(x) / ( 1 + tan(x)^2 ):  " = simplify(");
```

$$2\,\frac{\sin(x)}{1+\tan(x)^2} = 2\sin(x)\cos(x)^2$$

Ha a $\sin^2 x \longrightarrow 1 - \cos^2 x$ szabálynál a $\cos^2 x \longrightarrow 1 - \sin^2 x$-et jobban kedveljük, akkor mellékföltételekre vonatkozó egyszerűsítést kell végeznünk. Erről a 14.7. alfejezetben lesz részletesebben szó.

```
> sin(x)^3 + cos(x)^3;
```

$$\sin(x)^3 + \cos(x)^3$$

```
> simplify(");
```

$$\cos(x)^3 + \sin(x) - \sin(x)\cos(x)^2$$

```
> simplify( "", {cos(x)^2+sin(x)^2=1}, [sin(x),cos(x)] );
```

$$\cos(x)^3 + \sin(x) - \sin(x)\cos(x)^2$$

```
> simplify( """, {cos(x)^2+sin(x)^2=1}, [cos(x),sin(x)] );
```

$$\sin(x)^3 + \cos(x) - \cos(x)\sin(x)^2$$

Ha csak a trigonometrikus függvényekre vonatkozó egyszerűsítést kívánunk, adjuk meg a `trig` kulcsszót:

```
> 4^(1/2) - sin(x)^2 - 1;
```

$$\sqrt{4} - \sin(x)^2 - 1$$

```
> " = simplify( ", 'trig' );
```

$$\sqrt{4} - \sin(x)^2 - 1 = \sqrt{4} - 2 + \cos(x)^2$$

```
> "" = simplify("");
```

$$\sqrt{4} - \sin(x)^2 - 1 = \cos(x)^2$$

Megjegyezzük még, hogy a normalizálás a trigonometrikus egyszerűsítés előtt történik. Ennek következményeit mutatják az alábbi sorok:

```
> ( sin(x)^3 - 1) / ( sin(x)^2 - 1 );
```

$$\frac{\sin(x)^3 - 1}{\sin(x)^2 - 1}$$

```
> simplify( ", 'trig' );
```

$$-\frac{-\sin(x) - 2 + \cos(x)^2}{\sin(x) + 1}$$

```
> normal("");
```

$$\frac{\sin(x)^2+\sin(x)+1}{\sin(x)+1}$$

```
> map( simplify, """, 'trig' );
```

$$-\frac{-1+\sin(x)-\sin(x)\cos(x)^2}{\cos(x)^2}$$

• **Inverz trigonometrikus és hiperbolikus függvények**

A **simplify** figyelembe veszi az arc*trig*(*trig*) $\longrightarrow x$ transzformációt is, ahol *trig* = sin, cos, tan, sinh, cosh, tanh stb., amennyiben ez alkalmazható. Továbbá a **combine**-hoz hasonlóan a **simplify** is „ismer" a tangens inverzére vonatkozó bizonyos transzformációkat:

```
> simplify( arcsin( sin(x) ) );
```

$$\arcsin(\sin(x))$$

```
> simplify( ", assume=RealRange(-Pi/2,Pi/2) );
```

$$x$$

```
> simplify( "", 'arctrig', 'symbolic' );
```

$$x$$

```
> arctan(x) + arctan(1/x):  " = simplify(");
```

$$\arctan(x)+\arctan\left(\frac{1}{x}\right)=\frac{1}{2}\operatorname{csgn}(x)\,\pi$$

• **exp, ln**

Az exponenciális függvényre a **simplify** majdnem ugyanúgy működik, mint az **expand**, az $\exp x \exp y \longrightarrow \exp(x+y)$ és az $\dfrac{1}{\exp x} \longrightarrow \exp(-x)$ szabályok alkalmazásától eltekintve.

```
> exp(x) * exp(y):  " = simplify(");
```

$$e^x\,e^y=e^{(x+y)}$$

```
> expand( rhs(") );
```

$$e^x\,e^y$$

```
> exp(x)^2:  " = simplify(");
```

$$(e^x)^2=e^{(2\,x)}$$

```
> 1/exp(x):  " = simplify(");
```

$$\frac{1}{e^x}=e^{(-x)}$$

A természetes alapú logaritmus esetében a **simplify** először faktorizálja az argumentumot, majd a következő transzformációs szabályokat alkalmazza:

$$
\begin{aligned}
\ln(x^y) &\longrightarrow y\ln(x) \text{ pozitív } x\text{-re és valós } y\text{-ra,}\\
\ln(x^y) &\longrightarrow y\ln(-x) \text{ negatív } x\text{-re és páros egész } y\text{-ra,}\\
\ln(x^y) &\longrightarrow y\ln(x) \text{ negatív } x\text{-re és páratlan egész } y\text{-ra,}\\
\ln(x^y) &\longrightarrow y\ln(x) \text{ páratlan egész } x\text{-re,}\\
\ln(x^y) &\longrightarrow \frac{y}{2}\ln(x^2) \text{ páros egész } x\text{-re,}\\
\ln(x\,y) &\longrightarrow \ln(x)+\ln(y) \text{ pozitív } x\text{-re,}\\
\ln(x\,y) &\longrightarrow \ln(-x)+\ln(-y) \text{ negatív } x\text{-re,}\\
\ln(\exp(x)) &\longrightarrow x \text{ valós } x\text{-re,}\\
\ln(\mathrm{LambertW}(x)) &\longrightarrow \ln(x)+\mathrm{LambertW}(x) \text{ pozitív } x\text{-re.}
\end{aligned}
$$

Az $\ln(x\,y) \longrightarrow \ln(x)+\ln(y)$ és az $\ln(x^y) \longrightarrow y\ln(x)$ egyszerűsítések mindig kikényszeríthetők az `ln` és a `symbolic` kulcsszavak egyidejű használatával vagy további föltételekkel:

```
> simplify( ln(x^2), ln );
```

$$\ln(x^2)$$

```
> " = simplify( ", ln, 'symbolic' );
```

$$\ln(x^2) = 2\ln(x)$$

```
> "" = simplify( "", ln, assume=positive );
```

$$\ln(x^2) = 2\ln(x)$$

```
> """ = simplify( """, ln, assume=negative );
```

$$\ln(x^2) = 2\ln(-x)$$

- **Gyökök és hatványok**

A **simplify** a legtöbb hatványra ugyanazt végzi, mint az **expand**, azzal a fontos kivétellel, hogy fölhasználja az $x^y x^z \longrightarrow x^{y+z}$ szabályt, és amint az alábbiakból ki fog derülni, egyszerűsíti a törtkitevős hatványokat is. Másik fontos eltérés az **expand**-tól, amit az alábbi példákból is észrevehet az Olvasó, hogy a **simplify** nagyobb gondot fordít az alkalmazott transzformációk érvényességére:

```
> x^y * x^z:  " = simplify(");
```

$$x^y\,x^z = x^{(y+z)}$$

```
> expand( rhs(") );
```

$$x^y\,x^z$$

```
> simplify( (x^y)^z );
```

$$(x^y)^z$$

```
> " = simplify( ", 'power', 'symbolic' );
```

$$(x^y)^z = x^{(y\,z)}$$

```
> simplify( (x/y)^z );
```

$$\left(\frac{x}{y}\right)^z$$

```
> " = simplify( ", 'power', 'symbolic' );
```

$$\left(\frac{x}{y}\right)^z = x^z\, y^{(-z)}$$

```
> simplify( (-x)^y );
```

$$(-x)^y$$

```
> " = simplify( ", 'power', 'symbolic' );
```

$$(-x)^y = (-1)^y\, x^y$$

```
> (x^(1/2))^(y/2): " = simplify(");
```

$$(\sqrt{x})^{(1/2\,y)} = x^{(1/4\,y)}$$

```
> simplify( (x*y)^z );
```

$$(x\,y)^z$$

```
> " = simplify( ", 'power', 'symbolic' );
```

$$(x\,y)^z = x^z\, y^z$$

Törtkitevős hatványokra a **simplify** és az **expand** nagyon különbözik. Ebben az esetben a **simplify** ténylegesen a **radsimp** (radical **simp**lification) eljárásra támaszkodik. A **radsimp** eljárást olyan speciális kifejezések egyszerűsítésére szánják, amelyekben négyzetgyökök és más törtkitevős hatványok fordulnak elő. Az ilyen egyszerűsítések gyakran bonyolultak és időigényesek. Ha a **simplify** hívásában előfordul a `radical` kulcsszó, a Maple tudja, hogy csupán ilyen típusú egyszerűsítéseket kértünk:

```
> (2/27)^(1/3): " = simplify(");
```

$$\frac{1}{27}\, 2^{1/3}\, 27^{2/3} = \frac{1}{3}\, 2^{1/3}$$

Figyeljük meg a különbséget:

```
> (2/27)^(1/3): " = simplify( ", 'power' );
```

$$\frac{1}{27}\, 2^{1/3}\, 27^{2/3} = \frac{1}{27}\, 2^{1/3}\, 27^{2/3}$$

```
> (2/27)^(1/3): " = simplify( ", radical );
```

$$\frac{1}{27}\, 2^{1/3}\, 27^{2/3} = \frac{1}{3}\, 2^{1/3}$$

```
> (1-y^2)^(3/2) - (1-y^2)^(1/2): " = simplify(");
```

$$(1-y^2)^{3/2} - \sqrt{1-y^2} = -\sqrt{1-y^2}\, y^2$$

```
> (1-sin(x)^2)^(3/2) - (1-sin(x)^2)^(1/2);
```

$$(1-\sin(x)^2)^{3/2} - \sqrt{1-\sin(x)^2}$$

```
> simplify( ", radical );
```

$$-\sqrt{1-\sin(x)^2}\, \sin(x)^2$$

```
> "" = simplify( "" );
```

$$(1-\sin(x)^2)^{3/2} - \sqrt{1-\sin(x)^2} =$$
$$-\mathrm{csgn}(\cos(x))\cos(x) + \mathrm{csgn}(\cos(x))\cos(x)^3$$

A gyökök egyszerűsítését gondosabban végzi a Maple:

```
> (x^4)^(5/4): " = simplify(");
```

$$(x^4)^{5/4} = x^4\,(x^4)^{1/4}$$

```
> (x^4)^(5/4): " = simplify( ", radical, assume=positive );
```

$$(x^4)^{5/4} = x^5$$

• **Egyéb egyszerűsítések**

A Maple ismer további függvényekre (a Gamma, a Riemann ζ, a hipergeometrikus stb.) alkalmazható egyszerűsítési szabályokat is. Két példa:

```
> GAMMA(n+1/2)/GAMMA(n-1/2): " = simplify(");
```

$$\frac{\Gamma(n+\frac{1}{2})}{\Gamma(n-\frac{1}{2})} = n - \frac{1}{2}$$

```
> readlib( hypergeom ):
> hypergeom([-1,-3/2],[1/2],z^2/t^2);
```

$$\mathrm{hypergeom}([\frac{-3}{2}, -1], [\frac{1}{2}], \frac{z^2}{t^2})$$

```
> simplify(");
```

$$\frac{t^2 + 3\,z^2}{t^2}$$

Ha nem akarjuk az összes rendelkezésre álló egyszerűsítést alkalmazni, akkor a **simplify** eljárás hívásakor explicit módon meg kell neveznünk azokat a függvényeket, amelyekre az egyszerűsítést végre kell hajtani (mint a fenti példában, ahol a `radical` kulcsszót használtuk). Figyeljük meg, hogy ez éppen ellenkezője annak, ahogy a függvények kifejtését elnyomhattuk:

```
> exp(x)*exp(y) + cos(x)^2 + sin(x)^2;
```

$$e^x\,e^y + \cos(x)^2 + \sin(x)^2$$

```
> simplify(");
```

$$e^{(x+y)} + 1$$

```
> simplify( "", exp );
```

$$e^{(x+y)} + \cos(x)^2 + \sin(x)^2$$

```
> simplify( """, 'trig' );
```

$$e^x\,e^y + 1$$

14.5. A **convert** eljárás

A (hiperbolikus) trigonometrikus függvényeket és ezek inverzeit tartalmazó kifejezések a **convert** eljárással hozhatók explicit módon különféle formákra. Néhány példa:

- (Hiperbolikus) trigonometrikus függvények exponenciális alakra konvertálása és ennek fordítottja:

```
> cos(x): " = convert( ", exp );
```

$$\cos(x) = \frac{1}{2}e^{(I\,x)} + \frac{1}{2}\frac{1}{e^{(I\,x)}}$$

```
> map( convert, ", 'trig' );
```

$$\cos(x) = \frac{1}{2}\cos(x) + \frac{1}{2}I\sin(x) + \frac{1}{2}\frac{1}{\cos(x) + I\sin(x)}$$

```
> simplify(");
```

$$\cos(x) = \cos(x)$$

```
> cosh(x): " = convert( ", exp );
```

$$\cosh(x) = \frac{1}{2}e^{x} + \frac{1}{2}\frac{1}{e^{x}}$$

```
> map( convert, ", 'trig' );
```

$$\cosh(x) = \frac{1}{2}\cosh(x) + \frac{1}{2}\sinh(x) + \frac{1}{2}\frac{1}{\cosh(x) + \sinh(x)}$$

```
> simplify(");
```

$$\cosh(x) = \cosh(x)$$

Figyeljük meg a különbséget az

```
> exp(x+I*y): " = convert( ", 'trig' );
```

$$e^{(x+I\,y)} = (\cosh(x) + \sinh(x))\,(\cos(y) + I\sin(y))$$

és az

```
> exp(x+I*y): " = evalc(");
```

$$e^{(x+I\,y)} = e^{x}\cos(y) + I\,e^{x}\sin(y)$$

parancs eredménye között.

- Inverz (hiperbolikus) trigonometrikus függvények konvertálása logaritmusos kifejezésekre:

```
> arcsin(x): " = convert( ", ln );
```

$$\arcsin(x) = -I\ln(\sqrt{1-x^2} + I\,x)$$

```
> arcsinh(x): " = convert( ", ln );
```

$$\operatorname{arcsinh}(x) = \ln(x + \sqrt{x^2+1})$$

A (hiperbolikus) trigonometrikus függvények és inverzeik előző konverziói egy lépésben is végrehajthatók az `expln` kulcsszó alkalmazásával.

• Trigonometrikus függvények konvertálása csak a tangens függvényt tartalmazó formára:

```
> sin(x): " = convert( ", tan );
```

$$\sin(x) = 2\,\frac{\tan(\frac{1}{2}\,x)}{1+\tan(\frac{1}{2}\,x)^2}$$

```
> cos(x): " = convert( ", tan );
```

$$\cos(x) = \frac{1-\tan(\frac{1}{2}\,x)^2}{1+\tan(\frac{1}{2}\,x)^2}$$

```
> sin(x)/cos(x): " = convert( ", tan );
```

$$\frac{\sin(x)}{\cos(x)} = \tan(x)$$

• (Hiperbolikus) trigonometrikus függvények konvertálása csak a (hiperbolikus) szinusz és koszinusz függvényt tartalmazó formára:

```
> tan(x): " = convert( ", 'sincos' );
```

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

```
> tanh(x): " = convert( ", 'sincos' );
```

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

• Trigonometrikus függvények konverziója csak szinuszt és koszinuszt tartalmazó alakra, valamint hiperbolikus függvények konverziója csak exponenciális függvényeket tartalmazó formára:

```
> tan(x): " = convert( ", 'expsincos' );
```

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

```
> tanh(x): " = convert( ", 'expsincos' );
```

$$\tanh(x) = \frac{(e^x)^2-1}{(e^x)^2+1}$$

• Ezen kívül a Maple-ben elvégezhető a faktoriálisok és a binomiális együtthatók konvertálása Gamma függvényekké és fordítva:

```
> n!: " = convert( ", GAMMA );
```

$$n! = \Gamma(n+1)$$

```
> rhs(") = convert( rhs("), 'factorial' );
```

$$\Gamma(n+1) = \frac{(n+1)!}{n+1}$$

```
> lhs(") = expand( rhs(") );
```

$$\Gamma(n+1) = n!$$

```
> binomial(n,k):  " = convert( ", GAMMA );
```

$$\mathrm{binomial}(n,\, k) = \frac{\Gamma(n+1)}{\Gamma(k+1)\,\Gamma(n-k+1)}$$

```
> binomial(n,k):  " = convert( ", factorial );
```

$$\mathrm{binomial}(n,\, k) = \frac{n!}{k!\,(n-k)!}$$

```
> rhs(") = convert( rhs("), binomial );
```

$$\frac{n!}{k!\,(n-k)!} = \mathrm{binomial}(n,\, k)$$

```
> multinomial(n,a,b,c,d):  " = convert( ", GAMMA ):
```

$$\mathrm{multinomial}(n,\, a,\, b,\, c,\, d) = \frac{\Gamma(n+1)}{\Gamma(a+1)\,\Gamma(b+1)\,\Gamma(c+1)\,\Gamma(d+1)}$$

14.6. Trigonometrikus egyszerűsítés

A trigonometrikus kifejezések helyettesítésének speciális szerepe van a Maple-ben. A számítógépes algebrai rendszer egy különleges szolgáltatást is nyújt: egyszerű kifejezések esetében képes velük ekvivalens kifejezéseket javasolni. A megfelelő Maple eljárás neve **trigsubs**, ezt a könyvtárból be kell tölteni. Néhány példa:

```
> readlib(trigsubs):  # load library function
> trigsubs( sin(2*x) );
```

$$[\sin(2\,x),\, \sin(2\,x),\, 2\sin(x)\cos(x),\, \frac{1}{\csc(2\,x)},\, \frac{1}{\csc(2\,x)},\, 2\,\frac{\tan(x)}{1+\tan(x)^2},$$
$$-\frac{1}{2}\,I\,(e^{(2\,I\,x)} - e^{(-2\,I\,x)})]$$

```
> convert( trigsubs( tan(x)^2 ), 'set' );
```

$$\left\{4\,\frac{\tan(\frac{1}{2}\,x)^2}{(1-\tan(\frac{1}{2}\,x)^2)^2},\, \frac{1}{\cot(x)^2},\, 4\,\frac{\cot(\frac{1}{2}\,x)^2}{(\cot(\frac{1}{2}\,x)^2-1)^2},\, \frac{4}{(\cot(\frac{1}{2}\,x)-\tan(\frac{1}{2}\,x))^2},\right.$$
$$-\frac{(e^{(I\,x)} - e^{(-I\,x)})^2}{(e^{(I\,x)} + e^{(-I\,x)})^2},\, \tan(x)^2,\, \sec(x)^2-1,\, \frac{\sin(x)^2}{\cos(x)^2},\, \frac{\sin(2\,x)^2}{(1+\cos(2\,x))^2},$$
$$\left.\frac{(1-\cos(2\,x))^2}{\sin(2\,x)^2}\right\}$$

```
> trigsubs( sin(x) + sin(y) );
```

$$[2\sin(\frac{1}{2}\,x+\frac{1}{2}\,y)\cos(-\frac{1}{2}\,x+\frac{1}{2}\,y)]$$

Csodákat azért ne várjunk: a Maple **trigsubs** eljárása nem ismer föl sok ekvivalenciát. Amikor trigonometrikus kifejezések helyettesítésére van szükség, a **subs** helyettesítő eljárás alternatívájaként használhatjuk a **trigsubs**-ot is. Ez azért előnyős, mert az első argumentumként megadott trigonometrikus egyenlőséget le is ellenőrzi a tudásbázis alapján.

```
> trigsubs( cos(2*x) = cos(x)^2 - sin(x)^2,
>   cos(2*x) + sin(x)^2 + 1);
```

$$\cos(x)^2+1$$

A trigonometrikus egyszerűsítés elég fáradságos dolog, mivel számolás közben szükségessé válhat a különböző reprezentációk közti konverzió. Erre egyszerű példaként szolgál a $\tan(3x)$ kifejtése:

```
> expand( tan(3*x) );
```

$$\frac{3\tan(x)-\tan(x)^3}{1-3\tan(x)^2}$$

Próbáljuk meg visszanyerni az eredeti kifejezést. Azt várnánk, hogy ehhez elegendő a következő parancs:

```
> combine(");
```

$$\frac{-3\tan(x)+\tan(x)^3}{-1+3\tan(x)^2}$$

Sajnos, nem. Először szinuszok és koszinuszok segítségével kell fölírni a kifejezést:

```
> convert( ", 'sincos' );
```

$$\frac{-3\,\frac{\sin(x)}{\cos(x)}+\frac{\sin(x)^3}{\cos(x)^3}}{-1+3\,\frac{\sin(x)^2}{\cos(x)^2}}$$

Túl optimisták vagyunk, ha azt hisszük, hogy most már jöhet a **combine**:

```
> combine(");
```

$$\frac{3\sin(4\,x)+\sin(6\,x)+3\sin(2\,x)}{3\cos(2\,x)+1+3\cos(4\,x)+\cos(6\,x)}$$

Előbb még egyszerűsíteni kell a szinuszokból és koszinuszokból álló racionális kifejezést:

```
> normal("");
```

$$\frac{\sin(x)\,(-3\cos(x)^2+\sin(x)^2)}{\cos(x)\,(-\cos(x)^2+3\sin(x)^2)}$$

```
> combine(");
```

$$\frac{\sin(3\,x)}{\cos(3\,x)}$$

Végül az eredményt tangenssé kell konvertálni:

```
> convert( ", tan );
```

$$\tan(3\,x)$$

A Maple V Release 4 trigonometrikus egyszerűsítések közben figyelembe veszi a változók tulajdonságaira vonatkozó ismereteit is. Például

```
> cos( x + n*Pi );
```

$$\cos(x + n\,\pi)$$

```
> expand(");
```

$$\cos(x)\cos(n\,\pi) - \sin(x)\sin(n\,\pi)$$

```
> assume( n, integer );
> ";
```

$$\cos(x)\,(-1)^{n^{\sim}}$$

Ezt a részt egy olyan példával zárjuk, amelyből kitűnik, mennyire nehézkes lehet a trigonometrikus egyszerűsítések végrehajtása a gyakorlatban. A feladat a következő összeg kiszámítása:

```
> Sum( cos( omega*t - alpha - n*beta ), n=-N..N );
```

$$\sum_{n=-N}^{N} \cos(\omega\,t - \alpha - n\,\beta)$$

```
> value("):
```

a terjedelmes eredményt nem írattuk ki; először egyszerűsítsünk.

```
> simplify("):
```

Az output még nem sokkal rövidebb az előzőnél, ezért nem jelenítettük meg. De annyit már nyertünk vele, hogy a **combine**-t most alkalmazva már egyszerűsödik az eredmény:

```
> combine(");
```

$$\frac{1}{2}(-\sin(-\alpha - N\,\beta + \omega\,t) + \sin(-\alpha + N\,\beta + \omega\,t) \\ + \sin(N\,\beta + \beta + \omega\,t - \alpha) - \sin(-N\,\beta - \beta + \omega\,t - \alpha))/\sin(\beta)$$

Mivel $\omega t - \alpha$ minden tagban előfordul, érdemes helyettesíteni:

```
> algsubs( omega*t-alpha = zeta, " );
```

$$\frac{1}{2}(-\sin(-N\,\beta + \zeta) + \sin(N\,\beta + \zeta) + \sin(N\,\beta + \beta + \zeta) \\ - \sin(-N\,\beta - \beta + \zeta))/\sin(\beta)$$

Kifejtjük és újra egyszerűsítjük a kifejezést:

```
> simplify( expand(") );
```

$$\frac{\cos(\zeta)\,(\sin(N\,\beta) + \sin(N\,\beta)\cos(\beta) + \cos(N\,\beta)\sin(\beta))}{\sin(\beta)}$$

Ismét egyesítjük a trigonometrikus tagokat, de $\cos\zeta$-t még változatlanul hagyjuk. Ezt így is elérhetjük:

```
> map( combine, " );
```

$$\frac{\cos(\zeta)\,(\sin(N\,\beta)+\sin(N\,\beta+\beta))}{\sin(\beta)}$$

Az eredményül kapott formula már nem is olyan csúf. Nézzük meg, hogy a szinusz addíciós képlete segítene-e. Átmenetileg hagyjuk el $\cos\zeta$-t:

```
> coeff( ", cos(zeta) );
```

$$\frac{\sin(N\,\beta)+\sin(N\,\beta+\beta)}{\sin(\beta)}$$

Hívjuk segítségül a **trigsubs**-ot:

```
> readlib(trigsubs):
> subs( s=beta+N*beta, trigsubs( subs(
>     beta+N*beta=s, numer("") ) ) );
```

$$[2\sin(N\,\beta+\frac{1}{2}\,\beta)\cos(-\frac{1}{2}\,\beta)]$$

```
> trigsubs( denom(""") );
```

$$\left[\sin(\beta),\,\sin(\beta),\,2\sin(\frac{1}{2}\,\beta)\cos(\frac{1}{2}\,\beta),\,\frac{1}{\csc(\beta)},\,\frac{1}{\csc(\beta)},\,2\,\frac{\tan(\frac{1}{2}\,\beta)}{1+\tan(\frac{1}{2}\,\beta)^2},\right.$$
$$\left.-\frac{1}{2}\,I\,(e^{(I\,\beta)}-e^{(-I\,\beta)})\right]$$

A harmadik helyettesítési opció a legígéretesebb, mivel ekkor esik ki a legtöbb tag:

```
> op(1,"") / op(3,");
```

$$\frac{\sin(N\,\beta+\frac{1}{2}\,\beta)}{\sin(\frac{1}{2}\,\beta)}$$

A keresett összeg tehát

```
> subs( zeta=omega*t-alpha, cos(zeta)*" );
```

$$\frac{\cos(\omega\,t-\alpha)\sin(N\,\beta+\frac{1}{2}\,\beta)}{\sin(\frac{1}{2}\,\beta)}$$

Szép eredmény, de elég hosszú út vezetett idáig.

14.7. Mellékföltételekre vonatkozó egyszerűsítés

Tekintsük az 1991. szeptember 6-i Holland Matematikai Olimpia következő feladatát. Legyenek a, b és c olyan valós számok, amelyekre

$$a+b+c=3, \quad a^2+b^2+c^2=9, \quad a^3+b^3+c^3=24$$

teljesül. Számoljuk ki $a^4+b^4+c^4$ értékét.

A Maple megoldása a következő:

```
> siderels := { a+b+c=3, a^2+b^2+c^2=9, a^3+b^3+c^3=24 };
```

$$siderels := \{a^3+b^3+c^3=24,\ a+b+c=3,\ a^2+b^2+c^2=9\}$$

```
> simplify( a^4+b^4+c^4, siderels );
```

$$69$$

Annak megértéséhez, hogy a Maple hogyan számolta ki ezt az eredményt, némi fogalmunk kell, hogy legyen a *Gröbner bázisokról* és használatukról. Ebben a részben csak a Gröbner bázisok alapgondolatát ismertetjük; „matematikaibb" jellegű bevezetést találhatunk a [13, 26, 27, 51]-ben.

Az első lépésben a Maple veszi a mellékföltételeket megadó egyenletekben szereplő polinomok halmazát:

```
> polys := map( lhs - rhs, siderels );
```

$$polys := \{a^3+b^3+c^3-24,\ a+b+c-3,\ a^2+b^2+c^2-9\}$$

Ezután kiszámolja a valódi lexikografikus rendezésre vonatkozó minimális, főpolinomokból álló Gröbner bázist. Kicsit leegyszerűsítve a Gröbner bázis polinomok olyan halmaza, amely ugyanazt az ideált generálja, mint a kiindulási polinomok halmaza, és még bizonyos extra tulajdonságokkal is bír. A bázist kiszámoló Maple csomag neve `grobner`. A munkát elvégző csomagbeli eljárás neve **gbasis**:

```
> with( grobner ): # load the Groebner basis package
> G := gbasis( polys, [a,b,c], 'plex' );
```

$$G := [a+b+c-3,\ b^2+c^2-3\,b-3\,c+b\,c,\ 1-3\,c^2+c^3]$$

A Gröbner bázis függ az a, b, és c monomiálisainak rendezésétől. Itt az $a \succ b \succ c$ valódi lexikografikus rendezést használtuk. (V. ö. 5.2.)

A Gröbner bázisok jellemzése. *Polinomok egy véges G halmaza Gröbner bázis, ha a G által generált ideál minden eleme nullára redukálható a $\succ$ rendezésre vonatkozó „redukciós szabályok" alkalmazásával.*

A „redukció" jelentését legkönnyebben egy példával tudjuk megmagyarázni. Az első polinomból az a-ra vonatkozó redukcióval

$$a \longrightarrow 3-b-c$$

A második polinomnál a „legkönnyebb” valódi lexikografikus rendezésre vonatkozó redukció a „legnagyobb” monomiális eliminálása:

$$b^2 \longrightarrow -b\,c + 3b - c^2 + 3c$$

(bc a jobboldalon előforduló legnagyobb monomiális). A harmadik polinom „legkönnyebb” redukciója a legmagasabb fokszámú tag kiküszöbölése:

$$c^3 \longrightarrow 3c^2 - 1.$$

Ha ezeket a redukciókat addig alkalmazzuk egy polinomra, ameddig csak lehetséges, megkapjuk az illető polinom *normálformáját.* A G Gröbner bázist úgy is jellemezhetjük, hogy a G által generált ideál minden elemének a nulla az egyértelműen meghatározott normálformája. Más szavakkal, a G Gröbner bázisban teljesül az, hogy a G által generált ideál tetszőleges polinomjára a redukciós szabályokat alkalmazva nullát kapunk eredményül.

A G Gröbner bázist minimálisnak és monikusnak nevezzük, ha G minden g elemének főegyütthatója 1 és g a G/g-re nézve normálformájú. Minimális monikus Gröbner bázisra nézve bármely polinom normálformája egyértelmű. Ez „kanonikus alak” abban az értelemben, hogy két polinom ekvivalens, ha normálformájuk pontosan ugyanaz a polinom.

A Maple a **normalf** eljárást biztosítja a normálforma kiszámítására. Határozzuk meg $a^4+b^4+c^4$ normálformáját a kiszámított Gröbner bázisra vonatkozóan:

```
> normalf( a^4+b^4+c^4, G, [a,b,c], 'plex' );
```

$$69$$

Tehát a polinomiális mellékföltételekre vonatkozó egyszerűsítés nem más, mint a polinom normálformájának meghatározása a mellékföltételek által generált ideál valamely rendezéshez tartozó minimális redukált Gröbner bázisára vonatkozóan. Racionális kifejezések esetén a polinomiális mellékföltételekre vonatkozó egyszerűsítés a normalizált hányados számlálójára és nevezőjére külön-külön történik. Amennyiben a változókat nem specifikáltuk, vagy halmazként adtuk meg a határozatlanokat, a rendszer a teljes fokszám szerinti rendezést veszi. Ha a változókat listaként definiáltuk, az ezáltal indukált valódi lexikografikus rendezést veszi a rendszer. (Ezen fogalmak definícióit az 5.2. alfejezetben találjuk.) A mellékföltételekre vonatkozó egyszerűsítés végeredményének kontrollálására gyakran bölcs dolog a változókat a szükséges sorrendben megadni.

```
> simplify( x^3 + y^3, {x^2 + y^2 = 1}, [x,y] );
```

$$y^3 + x - x\,y^2$$

```
> simplify( x^3 + y^3, {x^2 + y^2 = 1}, [y,x] );
```

$$x^3 - y\,x^2 + y$$

```
> simplify( (x^3-y^3) / (x^3+y^3), {x^2 + y^2 = 1} );
```

$$\frac{x^3 + y\,x^2 - y}{x^3 - y\,x^2 + y}$$

Hasonlítsuk össze az alábbi egyszerűsítéssel:

```
> siderel := { cos(x)^2 + sin(x)^2 = 1 };
```

$$siderel := \{\cos(x)^2 + \sin(x)^2 = 1\}$$

```
> eqn := cos(x)^3 + sin(x)^3;
```

$$eqn := \cos(x)^3 + \sin(x)^3$$

```
> simplify( eqn, siderel, [cos(x),sin(x)] );
```

$$\sin(x)^3 + \cos(x) - \cos(x)\sin(x)^2$$

```
> simplify( eqn, siderel, [sin(x),cos(x)] );
```

$$\cos(x)^3 + \sin(x) - \sin(x)\cos(x)^2$$

A példából világosan kitűnik, hogyan alakítja át az általánosított racionális kifejezéseket a mellékföltételekkel meghatározott egyszerűsítés.

A mellékföltételekre vonatkozó egyszerűsítés rendkívül hatékony eszköz. Tekintsük például a következő f polinom egyszerűsítését:

```
> f ;
```

$$y^3 x^6 - 3y^3 x^4 + 3y^2 x^5 + 8y^2 x^4 + 3y^3 x^2 - 6y^2 x^3 + 3y x^4 - 16y^2 x^2 + 16y x^3 - y^3 + 3y^2 x + 23y x^2 + x^3 + 8y^2 - 16y x + 8x^2 - 26y + 26x + 40$$

```
> simplify( f, { u = x^2*y - y + x + 4 }, {x,y} );
```

$$u^3 - 4u^2 + 10u$$

Papírral és ceruzával nehéz az ilyen polinom-kompozíciókat megtalálni és ellenőrizni.

Mivel a Gröbner bázissal kapcsolatos számítások idő- és memóriaigénye óriásira nőhet, sokszor a **match** eljárás alkalmazása bizonyul jobb alternatívának, ez ugyanis polinomiális idejű algebrai mintaillesztéses algoritmust használ:

```
> guess := a*u^3 + b*u^2 + c*u + d:
> u := x^2*y - y + x + 4:
> match( f=guess, x, 'parms' );
```

true

```
> parms;
```

$$\{d = 0,\ a = 1,\ y = y,\ c = 10,\ b = -4\}$$

```
> subs( parms, eval(guess,1) );
```

$$u^3 - 4u^2 + 10u$$

Ennek kapcsán megemlítjük, hogy a Maple-ben polinomok kompozícióit a **compoly** eljárással határozhatjuk meg. A fönti f függvényre a **compoly** a

```
> compoly(f);
```

$$40 + 26y + 8y^2 + y^3,\ y = y x^2 - y + x$$

kompozíciót adja. Ezt úgy kell érteni, hogy $f = 40 + 26v + 8v^2 + v^3$, ahol $v = yx^2 - y + x$. Amint látható, a polinomok kompozíciója nem egyértelmű.

Másik példaként tekintsük a következő egyváltozós polinomot.

```
> f := x^6 + 6*x^4 + x^3 + 9*x^2 + 3*x - 5;
```

$$f := x^6 + 6\,x^4 + x^3 + 9\,x^2 + 3\,x - 5$$

```
> compoly(f);
```

$$x^2 - 5 + x,\ x = 3\,x + x^3$$

Azaz f a $g \circ h$ kompozícióval állítható elő, ahol $g = x^2 + x - 5$ és $h = x^3 + 3x$. Ellenőriztessük a Maple-lel az eredményt:

```
> subs( "[2], "[1] );
```

$$(3\,x + x^3)^2 - 5 + 3\,x + x^3$$

```
> expand( " - f );
```

$$0$$

Megjegyezzük, hogy ez a kompozíció sem egyértelmű: f fölírható $g \circ h$ alakban a $g = x^2 - \frac{21}{4}$ és a $h = x^3 + 3x + \frac{1}{2}$ polinomokkal is.

Az egyváltozós polinomokra alkalmazott algoritmus a [10, 92]-ben található meg. Az általánosabb esetet, egyváltozós racionális függvény előállítását racionális függvények kompozíciójaként, a [93, 94, 201] tárgyalja.

14.8. Az egyszerűsítés irányítása

Az egyszerűsítés folyamatát alapvetően három módon befolyásolhatjuk:

- föltételek megadásával,
- az érvényesség vizsgálatának elhagyásával,
- az alkalmazható transzformációk megszorításával függvények bizonyos osztályára.

A fejezet során már láttunk ilyen jellegű példákat. A mostani részben ezeket összegezzük, és még további példákat is hozunk.

• Egyszerűsítés föltételekkel

A **simplify**(*expression*, `assume` = *property*) parancs az *expression* kifejezést úgy egyszerűsíti, hogy föltételezi, hogy a benne szereplő változókra a *property*-ben megadott föltételek teljesülnek. Így olyan egyszerűsítések is végrehajthatók, amelyeknek egyébként semmilyen vagy csak részleges hatása lenne. Például

```
> expr := sqrt((x-1)^2);
```

$$expr := \sqrt{(x-1)^2}$$

```
> simplify( expr );
```

$$\mathrm{csgn}(x-1)\,(x-1)$$

```
> simplify( expr, assume=real );
```

$$\mathrm{signum}(x-1)\,(x-1)$$

```
> simplify( expr, assume=RealRange(1,infinity) );
```

$$x-1$$

```
> simplify( expr, assume=RealRange(-infinity,1) );
```

$$1-x$$

Figyeljük meg, hogy a `real` föltétel *nem* azt jelenti, hogy a számítások tartományát a valós számok halmazára szorítottuk meg, hanem csupán azt, hogy a változókról tettük föl, hogy valósak. Ez magyarázza azt is, hogy

```
> simplify( (-1)^(1/3), assume=real );
```

$$\frac{1}{2}+\frac{1}{2}I\sqrt{3}$$

miért ad még mindig komplex eredményt, holott ehelyett talán a -1-et vártuk volna.

Ha a változókra az **assume**-mal explicit kikötéseket teszünk, a Maple ezt az információt is hasznosítja (ha tudja) az egyszerűsítés során:

```
> ln( exp(x) );
```

$$\ln(e^x)$$

```
> assume( x, real ):
> ln( exp(x) );
```

$$x\tilde{}$$

```
> assume( x>0 ):
> (x^3*y)^(1/3):  " = simplify(");
```

$$(x\tilde{}^3\,y)^{1/3} = x\tilde{}\,y^{1/3}$$

```
> assume( y, RealRange(-Pi/2,Pi/2) ):
> arcsin(sin(y)):  " = simplify(");
```

$$\arcsin(\sin(y\tilde{})) = y\tilde{}$$

- **Az érvényesség vizsgálatának elhagyása**

Az egyszerűsítést végző eljárások hívásakor megadott `symbolic` kulcsszó azt jelenti, hogy bár a transzformáció helyessége nem bizonyítható, azért végre kell hajtani. A `delay` kulcsszó ellentétes hatású: ha nem látható be a transzformáció érvényessége, akkor kiértékeletlenül visszadja a kifejezést. Ebben az esetben lényeges különbség van a kiértékeletlenül visszadott kifejezés és az előjelfüggvénnyel kódolt válasz között. Ezt mutatja az alábbi példa:

```
> expr := (x^2)^(1/2); # no automatic simplification
```

$$expr := \sqrt{x^2}$$

```
> simplify( expr, 'symbolic' ); # simplify regardless
```

$$x$$

```
> simplify( expr, 'delay' ); # valid simplify
```

$$\sqrt{x^2}$$

```
> simplify( expr ); # simplify with sign functions
```

$$\mathrm{csgn}(x)\,x$$

• **Az egyszerűsítések megszorítása**

Nézzük a következő Maple fejtörőt:

```
> (4^x-1)/(2^x-1);
```

$$\frac{4^x-1}{2^x-1}$$

```
> simplify(");
```

$$\frac{4^x-1}{2^x-1}$$

```
> normal(");
```

$$\frac{4^x-1}{2^x-1}$$

```
> combine( ", 'power' );
```

$$\frac{4^x-1}{2^x-1}$$

Egyik egyszerűsítő eljárás sem tudja fölismerni, hogy 4^x valójában $(2^x)^2$, és nem találja meg a 2^x+1 egyszerűsített alakot. Ehhez először hatványokból exponenciális- és logaritmusfüggvényekké kell alakítani a kifejezést, majd speciális módon kell egyszerűsíteni:

```
> convert( ", exp );
```

$$\frac{e^{(x\ln(4))}-1}{e^{(x\ln(2))}-1}$$

```
> simplify( ", ln );
```

$$\frac{e^{(2\,x\ln(2))}-1}{e^{(x\ln(2))}-1}$$

```
> simplify( ", exp );
```

$$\frac{2^{(2\,x)}-1}{2^x-1}$$

```
> normal( ", 'expanded' );
```

$$2^x+1$$

A **combine**, **simplify** és a **convert** eljárások hívásakor második argumentumot megadva az egyszerűsítéseket egy bizonyos matematikai függvényre vagy kifejezések bizonyos osztályaira korlátozhatjuk. Az **expand** eljárás esetében a helyzet pontosan ennek ellenkezője: az extra argumentum itt arról tájékoztatja a Maple-t, hogy mely függvényeket vagy kifejezéseket kell változatlanul hagynia az egyszerűsítés közben:

```
> expr := ln(2*x)+sin(2*x);
```

$$\mathit{expr} := \ln(2\,x)+\sin(2\,x)$$

```
> expand( expr, ln );
```

$$\ln(2\,x) + 2\sin(x)\cos(x)$$

```
> expand(expr, sin );
```

$$\ln(2) + \ln(x) + \sin(2\,x)$$

```
> expr := expr^2;
```

$$expr := (\ln(2\,x) + \sin(2\,x))^2$$

```
> expand( expr, ln, sin );
```

$$\ln(2\,x)^2 + 2\ln(2\,x)\sin(2\,x) + \sin(2\,x)^2$$

Ha könnyebb megmondani, hogy mely függvényeket szabad kifejteni, mint azt, hogy melyeknek kell érintetlenül maradni, akkor használjuk a **frontend** eljárást:

```
> frontend( expand, [expr] );
```

$$\ln(2\,x)^2 + 2\ln(2\,x)\sin(2\,x) + \sin(2\,x)^2$$

```
> frontend( expand, [expr], [{},{sin(2*x)}] );
```

$$\ln(2\,x)^2 + 4\sin(x)\cos(x)\ln(2\,x) + 4\sin(x)^2\cos(x)^2$$

```
> frontend( expand, [expr], [{`+`,`*`,trig},{}] );
```

$$\ln(2\,x)^2 + 4\sin(x)\cos(x)\ln(2\,x) + 4\sin(x)^2\cos(x)^2$$

A **frontend** opcionális harmadik argumentuma két halmazból álló lista: az első halmazban fölsorolt típusneveket és a második halmazban megadott kifejezéseket nem kell „befagyasztani” az egyszerűsítés során (az alapértelmezés [{`+`, `*`}, { }]). Az utolsó parancsban tehát azt adtuk meg, hogy fejtsük úgy ki a kifejezést, hogy közben a trigonometrikus függvények kivételével a többi függvényhívás változatlan maradjon. A fölsorolt típusnevek közt lehetnek általunk definiált típusok is. Például

```
> # sine of 2x
> `type/t` := z -> evalb( z=sin(2*x) ):
> expr := sin(2*x) + sin(2*y) + sin(4*x);
```

$$expr := \sin(2\,x) + \sin(2\,y) + \sin(4\,x)$$

```
> frontend( expand, [expr], [{`+`,`*`,t},{}] );
```

$$2\sin(x)\cos(x) + \sin(2\,y) + \sin(4\,x)$$

```
> # trigonometric function applied to a sum
> `type/t` := trig(`+`);
```

$$type/t := \mathrm{trig}(+)$$

```
> expr := sin(2*x)+sin(3*x)+sin(x+y)+sin(u-v);
```

$$expr := \sin(2\,x) + \sin(3\,x) + \sin(x+y) + \sin(u-v)$$

```
> frontend( expand, [expr], [{`+`,t},{sin}] );
```

$$\sin(2\,x) + \sin(3\,x) + \sin(x)\cos(y) + \cos(x)\sin(y) + \sin(u)\cos(-v) + \cos(u)\sin(-v)$$

```
> # trigonometric function applied to a product
> 'type/t' := trig('*');
```

$$type/t := \mathrm{trig}(*)$$

```
> frontend( expand, [expr], [{'*',t},{sin}] );
```

$$2\sin(x)\cos(x) + 4\sin(x)\cos(x)^2 - \sin(x) + \sin(x+y) + \sin(u-v)$$

14.9. Saját egyszerűsítő rutinok definiálása

A 14.2. alfejezetben a parciális trigonometrikus kifejtésről szóló példában már érintettük a függvényekre vonatkozó „saját" kifejtési szabályok programozását. Mondjuk az **F** függvény esetében ehhez csupán az **expand/F** eljárást kell definiálnunk. Ezután valahányszor olyen kifejezésre alkalmazzuk az **expand**-ot, amely **F**-et is tartalmazza, a Maple **F** minden hívására az **expand/F**-ben definiált kifejtést alkalmazza. Additív **F**-re például:

```
> 'expand/F' := proc(x)
> local y,i:
> y := expand(x):
> if type(y,'+') then sum( F(op(i,y)), i=1..nops(y) ) fi
> end:
> f(p+q) + F(r+s): " = expand(");
```

$$\mathrm{f}(p+q) + \mathrm{F}(r+s) = \mathrm{f}(p+q) + \mathrm{F}(r) + \mathrm{F}(s)$$

```
> F(p*(q+r)): " = expand(");
```

$$\mathrm{F}(p\,(q+r)) = \mathrm{F}(p\,q) + \mathrm{F}(p\,r)$$

Ahogy az **expand**-ról szóló részben már megígértük, bemutatjuk a hiperbolikus tangenst kifejtő eljárás olyan átírt változatát, amely a többi trigonometrikus függvénnyel konzisztens eredményt ad. Tehát az **expand/tanh** eljárást fogjuk úgy átdefiniálni, hogy az **expand/tan**-ra hasonlítson, és az eredeti helyett mindig ezt alkalmazza a rendszer. Ez a módszer prototípusként szolgálhat arra is, hogyan vehetünk föl saját könyvtárunkba úgy eljárásokat, hogy mindig ezek hajtódjanak végre a megfelelő beépített könyvtári rutinok helyett.

Az **expand/tanh** implementálásához némi ihletet adhat a hozzá hasonló **expand/tan**:

```
> interface( verboseproc=3 ): # make Maple more verbose
> readlib( 'expand/tan' ): # load library routine
> print( 'expand/tan' );

proc(y)
local x, n, t, i, S, c, N, D, T;
    x := expand(y);
    if x <> y then RETURN(expand(tan(x))) fi;
    if type(x, '+') then
        n := nops(x);
        t := [op(x)];
        t := traperror([seq(tan(i), i = t)]);
        if t = 'singularity encountered' then
```

```
            RETURN(tan(convert(t, `+`)))
        elif t = lasterror then ERROR(lasterror)
        fi;
        t := expand(t);
        for i from 0 to n do
            c := combinat['choose'](t, i);
            S[i] := convert(map(convert, c, `*`), `+`);
            if 1 < irem(i, 4) then S[i] := -S[i] fi
        od;
        N := convert([seq(S[2*i - 1], i=1..iquo(n+1,2))],`+`);
        D := convert([seq(S[2*i], i = 0 .. iquo(n, 2))], `+`);
        N/D
    elif type(x, `*`) and type(op(1, x), integer) then
        n := abs(op(1, x));
        t := tan(x/n);
        for i from 0 to n do
            S[i] := binomial(n, i)*t^i;
            if 1 < irem(i, 4) then S[i] := -S[i] fi
        od;
        N := convert([seq(S[2*i - 1], i=1.. iquo(n+1,2))],`+`);
        D := convert([seq(S[2*i], i = 0 .. iquo(n, 2))], `+`);
        N/D
    else tan(x)
    fi
end
```

Első látásra elég rettenetes. A forráskód figyelmes átolvasása és néhány kifejtés megfigyelése után nyilvánvalóvá válik, hogy a Maple a

$$\tan(x_1 + x_2 + \ldots + x_n) = \frac{S_1 - S_3 + S_5 - S_7 + \ldots + (-1)^{\lfloor \frac{n-1}{2} \rfloor} S_{2\lfloor \frac{n-1}{2} \rfloor + 1}}{1 - S_2 + S_4 - S_6 + \ldots + (-1)^{\lfloor \frac{n}{2} \rfloor} S_{2\lfloor \frac{n}{2} \rfloor}}$$

szabályt alkalmazza, ahol S_i a $\tan(x_j)$ változók szimmetrikus polinomja:

$$\prod_{j=1}^{n} (t - \tan(x_j)) = \sum_{i=0}^{n} (-1)^i \, S_i \, t^{(n-i)}.$$

Nem nehéz belátni, hogy hasonló formulák igazak a hiperbolikus tangensre is:

$$\tanh(x_1 + x_2 + \ldots + x_n) = \frac{S_1 + S_3 + S_5 + S_7 + \ldots + S_{2\lfloor \frac{n-1}{2} \rfloor + 1}}{1 + S_2 + S_4 + S_6 + \ldots + S_{2\lfloor \frac{n}{2} \rfloor}}$$

ahol S_i a $\tanh(x_j)$ változóknak az alábbi egyenlettel definiált szimmetrikus polinomja:

$$\prod_{j=1}^{n} (t - \tanh(x_j)) = \sum_{i=0}^{n} (-1)^i \, S_i \, t^{(n-i)}.$$

A fönti kifejtési formulát az **expand/tan** eljárásnak megfelelő stílusban implementáltuk, az eredményül kapott kódot a `tanh` fájlban tároltuk. Ennek tartalma:

```
# Expand tanh(f(x))  where f(x) is a sum of products by
# repeatedly applying the following two transformations.
#
#                              tanh(x) + tanh(y)
#             tanh(x+y) = ---------------------
#                              1 + tanh(x) tanh(y)
# and
#                               tanh(x)
#              tanh(2*x) = 2 ------------
#                                       2
#                             1 + tanh(x)
#
# Note, the code doesn't apply these two rules recursively
# because that would generate a messy rational expression
# which would then have to be simplified. Instead, the fol-
# lowing identities are used
#                                S[1] + S[3] + S[5] + ...
# tanh(a[1]+a[2]+...+a[n])  = --------------------------  (*)
#                                 1 + S[2] + S[4] + ...
#
# where the S[i] are the symmetric polynomials in tanh(a[i])
#
# The code essentially writes down the formula (*) as a
# rational expression  in expand( tanh(a[i]) ).
# The result is not further expanded or normalized.
#
# > expand(tanh(x+y+z));
#
#    tanh(y) + tanh(z) + tanh(x) + tanh(y) tanh(z) tanh(x)
#  ---------------------------------------------------------
#   1 + tanh(y) tanh(z) + tanh(y) tanh(x) + tanh(z) tanh(x)
#
#
# > expand(tanh(2*x+2*y));
#
#                 tanh(x)          tanh(y)
#            2 ------------ + 2 ------------
#                         2                2
#               1 + tanh(x)      1 + tanh(y)
#           ----------------------------------
#                     tanh(x) tanh(y)
#           1 + 4 ---------------------------
#                            2               2
#                 (1 + tanh(x) ) (1 + tanh(y) )
#
# Author: Andre Heck
# Date:   Aug/95
# Remark: code is copied and adapted from expand/tan
`expand/tanh` := proc(y)
local x, n, t, i, S, c, N, D, T;
x := expand(y);
if x <> y then RETURN( expand(tanh(x)) ) fi;
if type(x,`+`) then
    n := nops(x);
    t := [op(x)];
    t := expand([seq(tanh(i),i=t)]);
```

```
        for i from 0 to n do
            c := combinat['choose'](t,i);
            S[i] := convert( map(convert,c,`*`), `+` );
        od;
        N := convert( [seq( S[2*i-1], i=1..iquo(n+1,2) )], `+` );
        D := convert( [seq( S[2*i], i=0..iquo(n,2) )], `+` );
        N/D # Don't expand the result
    elif type(x,`*`) and type(op(1,x),integer) then
        n := abs(op(1,x));
        t := tanh(x/n);
        for i from 0 to n do
            S[i] := binomial(n,i)*t^i;
        od;
        N := convert( [seq( S[2*i-1], i=1..iquo(n+1,2) )], `+`);
        D := convert( [seq( S[2*i], i=0..iquo(n,2) )], `+` );
        N/D # Don't expand the result
    else tanh(x)
    fi;
    end:
    savelib('`expand/tanh`','`expand/tanh.m`'):
    quit
```

Hol tároljuk ezt a fájlt? Utánozzuk a Maple könyvtárszerkezetét! A Maple könyvtár archivált formáját a `libname` környezeti változó által megadott alkönyvtárban a `maple.lib` nevű fájl tartalmazza. Unix platform esetén ennek szokásos helye:

```
> libname;
```

/usr/local/lib/maple/lib

Tegyük föl, hogy valóban itt találjuk a Maple könyvtárát. A Maple eljárások neve eredeti elhelyezkedésüket tükrözi a fájlrendszerben. Az **expand/tan** eljárást például eredetileg a `/usr/local/lib/maple/lib/expand/tan.m` fájl tartalmazza. Ha saját könyvtárunkat a `/home/user/private_maplelib` alkönyvtárban akarjuk létrehozni, a következő lépések végrehajtására van szükség. Ebben az alkönyvtárban hozzuk létre az `expand` alkönyvtárat és ennek `src` alkönyvtárát. A `/home/user/private_maplelib/expand/src` alkönyvtárban helyezzük el a fönti `tanh` fájlt. A következő rövid Maple szekciót ebben a könyvtárban futtasuk le:

```
> savelibname := `/home/user/private_maplelib`;
> read tanh
```

Ennek hatására a `/home/user/private_maplelib/expand` alkönyvtárban létrejön az **expand/tan** eljárást tartalmazó `tanh.m` fájl. Most már minden fölhasználásra kész állapotban van, legalábbis, ha saját könyvtárunk neve a **readlib** keresési listáján szereplő könyvtárnevek élén áll. Próbáljuk ki:

```
> libname := `/home/user/private_maplelib/`, libname:
> tanh(x+y+z):  " = expand(");
```

$$\tanh(x+y+z) = \frac{\tanh(x)+\tanh(y)+\tanh(z)+\tanh(x)\tanh(y)\tanh(z)}{1+\tanh(x)\tanh(y)+\tanh(x)\tanh(z)+\tanh(y)\tanh(z)}$$

```
> tanh(5*x):  " = expand(");
```

$$\tanh(5\,x) = \frac{5\tanh(x)+10\tanh(x)^3+\tanh(x)^5}{1+10\tanh(x)^2+5\tanh(x)^4}$$

14.10. Gyakorlatok

1. Mutassuk meg, hogy $\sum_{k=1}^{\infty} \frac{1}{k^2+1} = \frac{\pi}{2}\coth(\pi) - \frac{1}{2}$.

2. Fejtsük ki a $(\cos(2x)+1)^2$ kifejezést a $\cos^2(2x) + 2\cos(2x) + 1$ alakra, tehát úgy, mintha a $\cos(2x)$ polinomja lenne, s eközben a trigonometrikus függvényt ne fejtsük ki.

3. Vizsgáljuk meg, hogy a következő szimbolikus kifejezéspárok hogyan transzformálhatók egymásba a Maple segítségével.

(a) $x+y = \frac{1}{x+y}$ és $\frac{(x+y)^2+1}{x+y}$,

(b) $\exp(x+y)$ és $\exp(x)\exp(y)$,

(c) $\ln(x/y)$ és $\ln(x) - \ln(y)$,

(d) $x^{(y+z)}$ és $x^y x^z$,

(e) $\sqrt{x^2-1}$ és $\sqrt{x-1}\sqrt{x+1}$.

4. Egyszerűsítsük a következő szimbolikus kifejezéseket:

(a) $\frac{e^x + x}{e^{2x} + 2xe^x + x^2}$

(b) $\sqrt[3]{x^5 + 40x^4 + 595x^3 + 3905x^2 + 9680x + 1331}$

(c) $\frac{(x-2)^{3/2}}{(x^2-4x+4)^{1/4}}$

(d) $\frac{\sqrt{x}-y}{x-y^2}$

(e) $\frac{1}{2+5^{1/3}}$

(f) $\cos(x+y) + \sin x \sin y + 2^{x+y}$

(g) $2\cos^2 x - \cos 2x$

5. Oldjuk meg a következő zérusekvivalencia problémákat a Maple-lel:

(a) $(2^{1/3} + 4^{1/3})^3 - 6(2^{1/3} + 4^{1/3}) - 6 = 0$

(b) $\ln\,\tan(\frac{1}{2}x + \frac{1}{4}\pi) - \operatorname{arcsinh}\,\tan x = 0$

6. Ellenőrizzük a Maple segítségével a következő trigonometrikus azonosságokat:

(a) $\sin x + \sin y + 2\,\sin\,\frac{1}{2}(x+y)\,\cos\,\frac{1}{2}(x-y)$

(b) $\sin 5x = 5\sin x - 20\sin^3 x + 16\sin^5 x$

(c) $\cot^2 x + 1 = \csc^2 x$

(d) $\tan x + \tan y = \dfrac{\sin(x+y)}{\cos x \cos y}$

(e) $\cos^6 x + \sin^6 x = 1 - 3\sin^2 x \cos^2 x$

(f) $\sinh 2x = 2\dfrac{\tanh x}{1 - \tanh^2 x}$

(g) $\dfrac{\sin 2x + \sin 2y}{\cos 2x + \cos 2y} = \tan(x+y)$

7. Ellenőrizzük a Maple segítségével a $\pi/4 = 4\arctan(1/5) - \arctan(1/239)$ egyenlőséget.

8. Számoljuk ki a következő határozatlan integrálokat, majd ellenőrizzük az eredményt differenciálással és egyszerűsítéssel:

(a) $\displaystyle\int \frac{2}{\sqrt{4+x^2}}\,dx$

(b) $\displaystyle\int \sqrt{(1-cx^2)^3}\,dx$

(c) $\displaystyle\int \frac{1}{x^4-1}\,dx$

(d) $\displaystyle\int \frac{1}{x^4-4}\,dx$

(e) $\displaystyle\int \sin 3x \cos 2x\,dx$

9. Integráljuk az $x \mapsto \dfrac{1}{(ax+b)^2(cx+d)^2}$ valós függvényt, és hozzuk az eredményt a következő alakra:

$$\frac{2ac\ln\left(\dfrac{cx+d}{ax+b}\right)}{(ad-bc)^3} - \frac{2acx+ad+bc}{(ad-bc)^2(ax+b)(cx+d)}$$

10. Implementáljuk az **expand/coth** expanziós rutint az **expand/coth**-hoz hasonlóan, vagyis úgy, hogy az **expand** a hiperbolikus kotangenseket nem a hiperbolikus szinuszok és koszinuszok, hanem maguk a hiperbolikus kotangensek segítségével fejtse ki. Rendezzük el úgy a dolgokat, hogy a rendszer mindig saját expanziós rutinunkat használja a beépített helyett.

14.11. Egyszerűsítési táblázat

Eljárás	Trig. függvények	exp és log	Hatványok	Speciális függvények
expand	$\cos(x+y) \to \cos x \cos y - \sin x \sin y$ $\cos 2x \to 2\cos^2 x - 1$ $\cosh 3x \to 4\cosh^3 x - 3\cosh x$	$\exp(x+y) \to \exp x \exp y$ $\ln(xy) \to \ln x + \ln y$ $\ln(x/y) \to \ln x - \ln y$	$x^{(y+z)} \to x^y x^z$ $(xy)^z \to x^z y^z$ $(x/y)^z \to x^z (1/y)^z$	$(n+1)! \to (n+1)n!$ $\Gamma(n+\frac{3}{2}) \to (n+\frac{1}{2})\Gamma(n+\frac{1}{2})$ $-\mathrm{dilog}(\frac{1}{x}) \to \mathrm{dilog}(x) + \frac{1}{2}\ln^2 x$
combine	$\cos x \cos y - \sin x \sin y \to \cos(x+y)$ $2\sinh x \cosh x \to \sinh 2x$ $4\cos^3 x \to \cos 3x + 3\cos x$	$\exp x \exp y \to \exp(x+y)$ $\ln x + \ln y \to \ln(xy)$ $y \ln x \to \ln(x^y)$	$x^y x^z \to x^{(y+z)}$ $(x^y)^z \to x^{yz}$ $\sqrt{x+1}\sqrt{x} \to \sqrt{x^2+x}$	$\sum_k a_k + \sum_k b_k \to \sum_k (a_k + b_k)$ $\int f + \int g \to \int f + g$
simplify	$\cos^2 x + \sin^2 x \to 1$ $\cosh^2 x - \sinh^2 x \to 1$ $\tan x \to \sin x / \cos x$	$\exp x \exp y \to \exp(x+y)$ $\ln x + \ln y \to \ln(xy)$ $\ln(x^y) \to y \ln x$	$x^y x^z \to x^{(y+z)}$ $(x/y)^z \to x^z y^{-z}$ $\sqrt{x^2+2x+1} \to x+1$	$\Gamma(n+\frac{1}{2})/\Gamma(n-\frac{1}{2}) \to n - \frac{1}{2}$ ${}_2F_1\left(\frac{1-n}{2}, \frac{-n}{2}, \frac{1}{2}; \frac{z^2}{t^2}\right) \to$ $((t+z)^n + (t-z)^n)/2t^n$
convert	$\cos x \to (e^{ix} + e^{-ix})/2$ $\mathrm{arcsinh} x \to \ln(x + \sqrt{x^2+1})$ $\sin 2x \to 2\tan x/(1+\tan^2 x)$	$e^{ix} \leftrightarrow \cos x + i \sin x$	$\sqrt{a} \leftrightarrow \mathrm{RootOf}(_Z^2 - a)$	$\binom{n}{k} \to \frac{n!}{k!(n-k)!}$ és más típuskonverziók

14.1. táblázat: A legfontosabb egyszerűsítések

15. Grafika

A kétdimenziós grafika a következő területeket öleli föl:

- egyváltozós valós függvényekkel definiált görbék,
- paraméteres egyenletekkel definiált görbék,
- implicit módon, egyenlettel megadott görbék,
- kétváltozós valós függvényekhez, valamint adathalmazokhoz tartozó sűrűségi és szintvonalas ábrák,
- vektor- és gradiensmezők ábrázolása,
- (statisztikai) adatok megjelenítése,
- lineáris egyenlőtlenségekkel definiált régiók ábrázolása,
- geometriai ábrák (például poligonok, körök, ellipszisek stb.) és végül
- kétdimenziós grafikus objektumok animációja.

A Maple által biztosított háromdimenziós grafikával lehetőségünk van

- kétváltozós valós függvénnyel definiált felület generálására,
- paraméteres egyenletekkel definiált felületek, csövek és térgörbék generálására,
- egyenlettel megadott implicit felületek generálására,
- háromdimenziós adatpontok listájával megadott felületek generálására,

- szabályos poliéderek, hengerek, gömbök, tóruszok és hasonló geometriai objektumok ábrázolására, végül
- háromdimenziós grafikai objektumok animációjára.

Két- vagy háromdimenziós ábrák készítésekor a Maple dönti el a mintapontok számát, a tengelyek elhelyezését, a tengelyeken található jelzéseket, az ábrázolási tartományokat, az ábra árnyékolását vagy színezését és így tovább. A grafikonokat opciók használatával módosíthatjuk; például másik (polár, elliptikus, logaritmikus, szférikus, henger, paraboloid stb.) koordinátarendszer választásával vagy a mintapontok rácsának megváltoztatásával.

A függvények, illetve felületek ábrázolására szolgáló **plot** és **plot3d** eljárások hívása elég nyilvánvaló módon történik. További grafikai lehetőségek a `plots` csomaggal érhetők el.

A fejezet további részében föltesszük, hogy minden mintaként közölt Maple szekció elején betöltöttük a `plots` *csomagot.*

Erre szolgál a következő parancs:

```
> with(plots); # load graphics package
```

[*animate*, *animate3d*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *odeplot*, *pareto*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedraplot*, *replot*, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

Az itt fölsorolt nevek alapján már lehet némi elképzelésünk a Maple-ben könnyen elérhető két- és háromdimenziós grafikai specialitásokról. A **display** a `plots` csomag legfontosabb parancsa. Az ábrák különböző opcióknak megfelelő újrarajzolásán túl lehetővé teszi több ábra összekombinálását is.

Ebben a fejezetben a Maple V Release 4 X Window rendszer alatti változatának grafikai lehetőségeit írjuk le. A legtöbb grafikai példa azonban más fölhasználói felületek esetében is érvényes. Nem tárgyaljuk az összes lehetséges grafikai rutint és opciót, de súlyt helyezünk arra, hogy megértsük a rajzoló rutinoknál használt grafikai struktúrákat.

A mintaként közölt szekciók parancsaiban gyakran megadunk a színezésre vonatkozó információkat is, a könyv azonban fekete-fehérben tartalmazza az ezeknek megfelelő ábrákat.[1]

[1] Pontosabban az alkalmazott nyomdatechnikának megfelelő élességű és számú szürkeárnyalatot láthatunk. Az eredeti ábráktól való esetleges eltérések a hardver és szoftver differenciák mellett ennek számlájára is írhatók. (*A Fordító megjegyzése.*)

Ha az eredeti színeket szeretnénk látni, futtassuk le újra a példákat. Egyébként is a Maple grafikai lehetőségeivel való kísérletezésre szeretnénk buzdítani az Olvasót.

A fejezet néhány példája meglehetősen időigényes, és komoly számítógépes erőforrásokat köt le.

15.1. A kétdimenziós grafika alapjai

Egyváltozós függvényeket a Maple **plot** eljárásával ábrázolhatunk. A rendszernek természetesen tudnia kell, hogy milyen típusú output eszközt használunk. Ha a Maple munkalapos felületű változatát Macintosh-on, PC kompatibilis vagy Unix-os gépen futtatjuk, a megjelenítő eszköz és a megfelelő meghajtók kiválasztása automatikus. Egyébként meg kell adnunk a szöveges üzemmódból grafikus módba váltáshoz kiküldendő speciális karaktersorozatokat és más szükséges információkat a Maple-nek.

Ha a **plot** eljárást például Tektronix emulációval akarjuk használni MS-DOS gépen az MS-KERMIT alatt, ezt a

```
> plotsetup( tek, kermit, preplot=[27,12], postplot=[24] ):
```

paranccsal tehetjük meg. Az utasítás a következőket jelenti:

- tudatjuk a Maple-lel, hogy Tektronix 4014 terminált vagy annak megfelelő emulátort használunk;
- a `preplot` interfész változó értéke az `escape` és a `formfeed` karakterek ASCII kódjának megfelelő egészek listája; a rajzolás megkezdése előtt ezeket kell kiküldeni a grafikus módba történő átmenethez;
- a `postplot` interfész változó értéke a `cancel` karakter ASCII kódja; a rajzolás befejezése és az ENTER billentyű lenyomása után ezzel térünk vissza karakteres módba.

Néha a szükséges beállítások gyorsan elvégezhetők a **plotsetup** eljárás segítségével. A rendelkezésre álló fölhasználói felületek részleteiről a Maple kézikönyvekben (v. ö. [38, 39, 187, 188]) találunk információt.

A továbbiakban föltételezzük, hogy a Maple az X Window rendszer alatt fut, mivel ennek grafikus meghajtója alkalmas háromdimenziós ábrák előállítására is.

Tekintsük az $f : x \mapsto e^{-x^2}\sin(\pi x^3)$ függvényt a $(-2, 2)$ intervallumon.

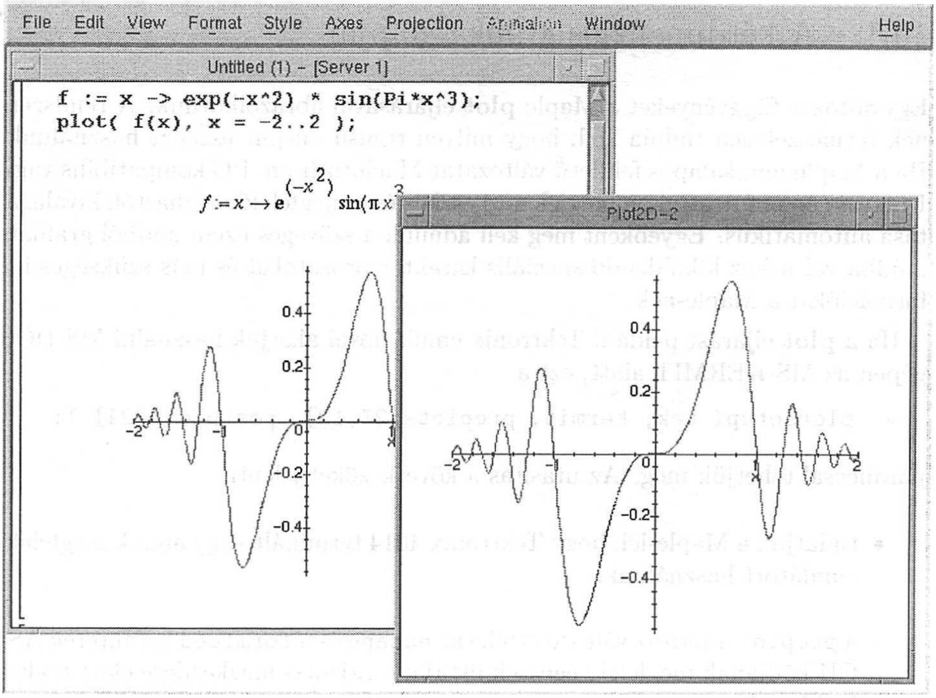
```
> f := x -> exp(-x^2) * sin(Pi*x^3);
```

$$f := x \to e^{(-x^2)}\sin(\pi\, x^3)$$

A függvény kirajzolását a

```
> plot( f(x), x = -2..2 );
```

paranccsal végezhetjük. Hatására kétdimenziós („PLOT2D") grafikát tartalmazó új ablak jelenik meg a képernyőn, amely a függvény grafikonját ábrázolja. A hozzá tartozó menü lehetőséget ad a Maple grafikai adatstruktúrán különböző manipulációk elvégzésére és az eredmény megjelenítésére.



15.1. ábra: Munkalapot és grafikai ablakot tartalmazó képernyőkép

Lehetséges, hogy a 15.1. ábrához hasonlóan a grafikont a munkalapba beágyazva látjuk (ténylegesen ez a rendszer alapföltételezés szerinti megjelenítési módja). A fenti utasítás példát ad az f *függvényt* az (a, b) intervallumon kirajzoló

$$\textbf{plot}(\, f,\ a..b,\ \textit{options});$$

általános alakú parancs használatára, ahol az *options* rész a következő alfejezetben tárgyalt opciókból fölépülő listát jelent, amely lehet üres is.

A másik módszer az $f(x)$ *formula* által definiált függvény ábrájának megjelenítésére a **plot** következő változata:

$$\textbf{plot}(\, f(x),\ x = a..b,\ \textit{options});$$

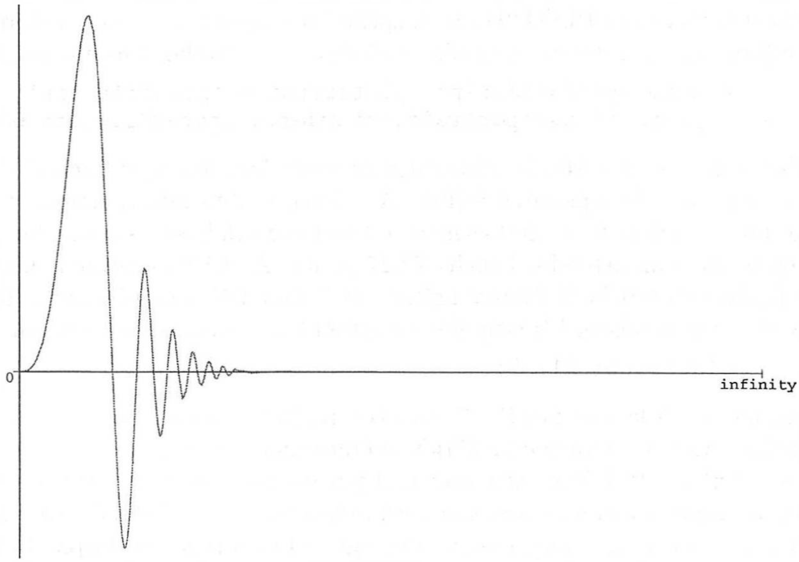
ahol az *a..b* a vízszintes tartomány (x-re), az *options* pedig nulla vagy több opciót tartalmaz. Figyeljük meg a különbséget a vízszintes tartomány meghatározásában *függvény*, illetve *formula* esetén: a függvény csak egy *tartományt* követel meg, míg a formuláknál a szintaxis *variable* = *range*.

Tehát az $e^{-x^2}\sin(\pi x^3)$ formulát a $(-2, 2)$ tartományba eső x értékekre a következő paranccsal rajzoltathatjuk föl:

```
> plot( f(x), x = -2..2 );
```

Az eredmény a 15.1. ábrától csak az X tengely mellé írt cimkében különbözik. A függvény grafikonját megvizsgálhatjuk a teljes valós számegyenesen vagy egy félegyenesen is. A 15.2. ábra az f függvényt mutatja a $(0, \infty)$ félegyenesen.

```
> plot( f, 0 .. infinity );
```

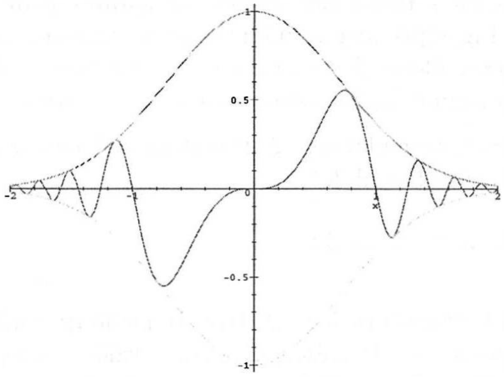


15.2. ábra: Végtelen intervallum fölötti ábrázolás

Ebben az esetben a Maple az $x \mapsto \frac{2}{\pi}\arctan\left(\frac{x}{2\pi}\right)$ közelítő függvény segítségével képezi le a teljes valós egyenest a $(-1, 1)$ intervallumba.

Egyszerre több függvényt is kirajzoltathatunk; színes megjelenítés esetén a Maple az egyes grafikus objektumokat eltérő színekkel ábrázolja:

```
> plot( {f(x), exp(-x^2), -exp(-x^2)}, x=-2..2 );
```



15.3. ábra: Az $x \mapsto e^{-x^2}\sin(\pi x^3)$, az e^{-x^2} és a $-e^{-x^2}$ függvények grafikonja

Ha ábránkat ki akarjuk nyomtatni, vagy PostScript formátumú fájlban akarjuk elhelyezni, akkor ezt többféleképpen közölhetjük a Maple-lel. Használhatjuk az ábrát tartalmazó PLOT2D ablak „print" menüpontját, vagy ha előre pontosan tudjuk, mit szeretnénk, megadhatunk egy, az alábbihoz hasonló parancsot:

```
> plotsetup( PostScript, plotoutput = 'plotfile.eps',
>   plotoptions='portrait,noborder,height=200,width=600' ):
```

Ennek hatására a Maple átirányítja az ábrát leíró Encapsulated PostScript kódot a `plotfile.eps` nevű fájlba. A keletkező ábra álló (portrait) formátumú, keretezés nélküli 600 × 200-as méretű lesz (a `width` hosszúság és a `height` szélesség pontokban értendő, 1 inch=72.27 pont). Az EPS szabványnak megfelelően a fájlban szerepel a `% BoundingBox: 0 0 600 200` méretdefiníció. Ha ismét az X Window rendszer képernyőjén szeretnénk ábráinkat megjeleníteni, adjuk ki a

```
> plotsetup( X11 ):
```

parancsot. A kapott fájl PostScript kompatibilis nyomtatón egyszerűen kinyomtatható vagy beágyazható a LaTeX, a Framemaker, esetleg más szövegszedő rendszerrel készített dokumentumokba. A parancsban fölsorolt opciók megadása nélkül a Maple olyan maximális méretű bekeretezett fekvő (landscape) formátumú képet generál, ami még éppen elfér egy A4-es lapon. A Maple által támogatott további plot eszközök: `win` (MS-Windows), a plotterekben is használt `hpgl` (Hewlett–Packard Graphics Library) és `gif` (GIF formátum). További részleteket a **?plot,device** paranccsal tudhatunk meg.

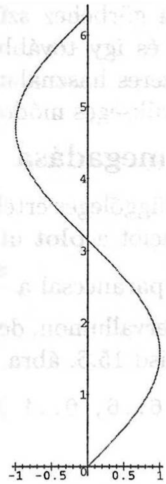
A Descartes-féle koordinátarendszerben az $x = f(t)$, $y = g(t)$ *paraméteres formulákkal* adott kétdimenziós síkgörbéket kirajzoló utasítás általános alakja

$$\mathbf{plot}(\,[\,f(t),\ g(t)\ t = a..b\,],\ \mathit{options});$$

ahol a t független változó az $a..b$ tartományból vesz föl értékeket, az *options* pedig opciók (esetleg üres) listája.

A 15.4. ábrán egyszerű példát láthatunk az előzőekre:

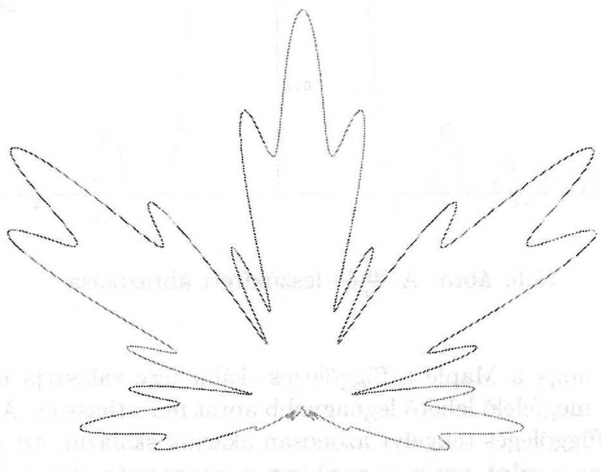
```
> plot( [ sin(t), t, t=0..2*Pi ], scaling=constrained );
```



15.4. ábra: A $t \mapsto (\sin t, t)$ paraméteres görbe rajza

Ha a paraméteres görbét *függvényekkel* adtuk meg, a **plot** parancsból el kell hagynunk a független változót. A 15.5. ábrán egy érdekes példát láthatunk: függvényekkel megadott, juharlevélhez hasonló alakú paraméteres görbét. Az előállításban polárkoordináta-rendszert használtunk.

```
> S := t -> 100/(100+(t-Pi/2)^8): # for scaling
> R := t -> S(t)*(2-sin(7*t)-cos(30*t)/2):
> plot( [ R, t->t, -Pi/2..3/2*Pi ], coords=polar,
> axes=none, color=green, numpoints=1000 );
```



15.5. ábra: Paraméteres görbével előállított juharlevél

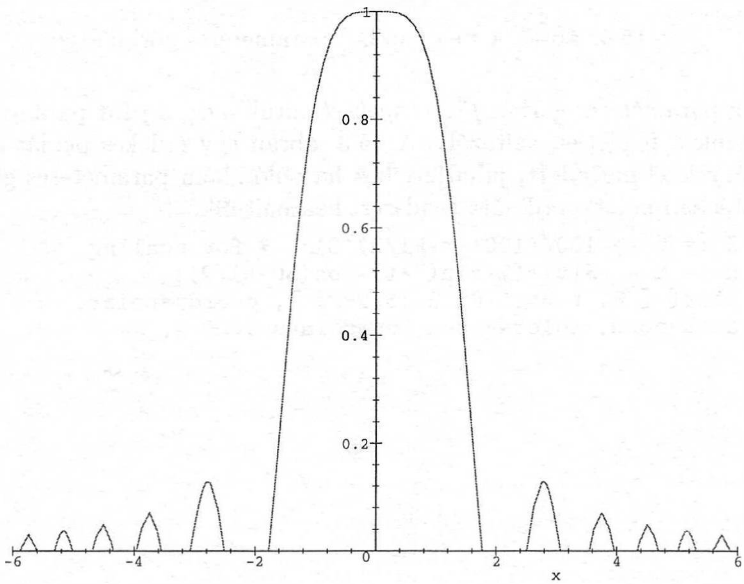
15.2. A plot opciói

Amikor a Maple egy ábrát fölrajzol, számos döntést hoz. Meghatározza például az ábrázolási tartományt, a síma görbéhez szükséges mintapontok számát, a tengelyek beosztását és föliratát és így tovább. A Maple gondoskodik arról, hogy ezen opciók értéke a rendszeres használatra a legalkalmasabb legyen, de beállíthatjuk őket a számunkra szükséges módon is.

A függőleges tartomány megadása

Ha például a grafikonon ábrázolt függőleges értéktartományt akarjuk korlátozni, a tartományra vonatkozó információt a **plot** utasítás harmadik argumentumaként adhatjuk meg. A következő paranccsal a $\dfrac{\sin x^2}{x^2}$ formulával definiált függvényt rajzoltatjuk föl a $(-6, 6)$ intervallumon, de a $[0,\ 1]$ függőleges tartományon kívül eső értékeket elhagyjuk. (Lásd 15.6. ábra.)

```
> plot( sin(x^2)/x^2, x=-6..6, 0..1 );
```

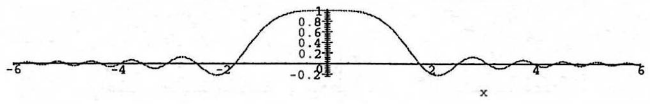


15.6. ábra: A $\frac{\sin x^2}{x^2}$ leszűkített ábrázolása

Skálázás

Figyeljük meg, hogy a Maple a függőleges skálát úgy választja meg, hogy az A4-es méretnek megfelelő lehető legnagyobb ábrát mutathasson. Amennyiben a vízszintes és a függőleges tengelyt azonosan akarjuk skálázni, ezt megadhatjuk interaktívan vagy a **plot** parancs `scaling = constrained` opciójával. (Lásd 15.7. ábra.)

```
> plot( sin(x^2)/x^2, x=-6..6, scaling=constrained );
```

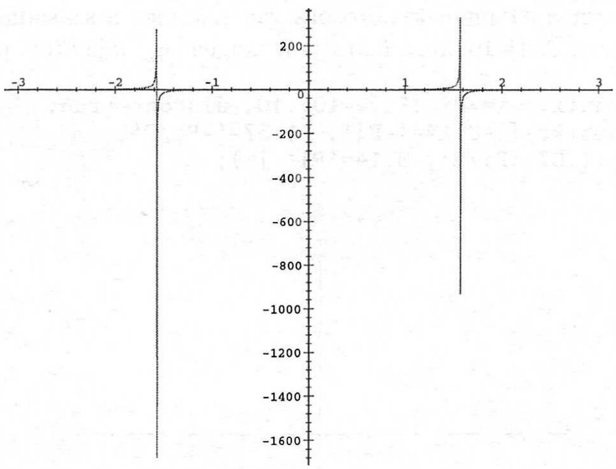


15.7. ábra: A $\frac{\sin^2 x}{x^2}$ leszűkített ábrázolása egyenlő skálázással

A megfelelő nézet

Néha szűkítenünk kell a függőleges tartományt ahhoz, hogy jó ábrát kapjunk. A 15.8. ábra grafikonján látható, hogy néhány függvényérték dominál a többi fölött, ezért a tangens függvény menetét az ábra nem túl jól tükrözi az adott tartományon. A (hamis) „tüskék" elkerülésének egyetlen módja sokszor a függőleges tartomány ésszerű szűkítése vagy a mintapontok számának növelése.

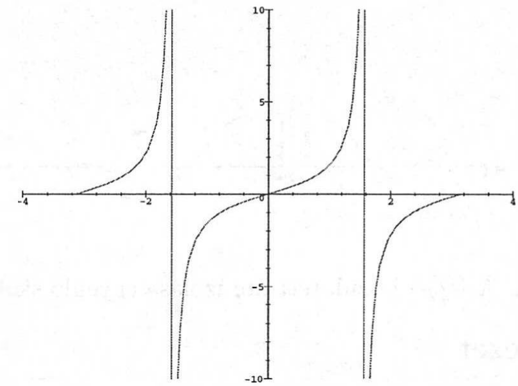
```
> plot( tan, -Pi..Pi );
```



15.8. ábra: A hibás tangensgörbe

A következő paranccsal újrarajzoltatjuk a $[-\pi, \pi]$ fölötti tangens függvényt, de a $[-10, 10]$ függőleges tartományon kívül eső értékeket nem mutatjuk meg, a vízszintes tartományt pedig a $[-4, 4]$-re bővítjük. A rendszer nem számolja ki újra a grafikont, csak a megjelenítést igazítja hozzá ehhez a specifikációhoz:

```
> display( ", view=[-4..4,-10..10] );
```

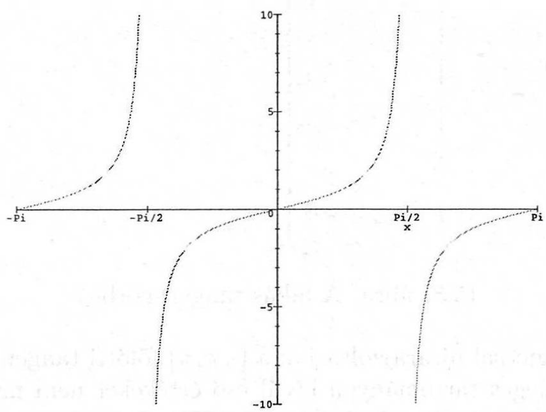


15.9. ábra: Az újrarajzolt tangensgörbe

Szakadási helyek

A Maple valójában az $x = -\frac{\pi}{2}$-nél és az $x = \frac{\pi}{2}$-nél lévő függőleges aszimptotákat is a görbe részeként rajzolta be, mivel a szakadásokkal nem törődve a nagyon nagy és a nagyon kicsi függvényértékeket adó mintapontokat egyenessel kötötte össze. A `discont = true` opcióval arra kérhetjük a Maple-t, hogy próbálja meg a szakadásokkal kapcsolatos bonyodalmakat legyőzni. Ekkor kifejezésekkel kell dolgoznunk, mert a Maple-nek változóra van szüksége a szakadások helyének megállapításához. A 15.10. ábra mutatja a tangens így kijavított grafikonját:

```
> plot( tan(x), x=-Pi..Pi, -10..10, discont=true,
>   xtickmarks=[ -3.14='-Pi', -1.57='-Pi/2',
>         1.57='Pi/2', 3.14='Pi' ] );
```

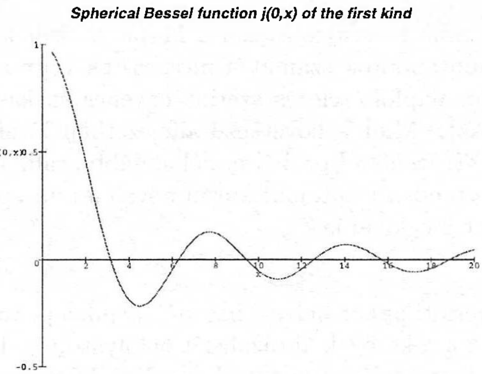


15.10. ábra: A tangensgörbe javított ábrázolása

Címek, cimkék, osztásjelek

Az előző ábra vízszintes tengelyére az `xtickmarks` opcióval helyeztünk el értelmesebb cimkéket. Természetesen létezik az `ytickmarks` opció is. A cimkék explicit megadása helyett elegendő a két tengelyen lévő osztásjelek számát előírni. A grafikonokon cimkéket helyezhetünk el, megmondhatjuk azt is, hogy milyen fontokat szeretnénk használni (lásd 15.11. ábra). Ha a cimkézést a fönti módon végezzük, a cimkék elhelyezésére nincs sok befolyásunk; legföljebb utólag, a PostScript kód megfelelő átírásával végezhetünk korrekciókat.

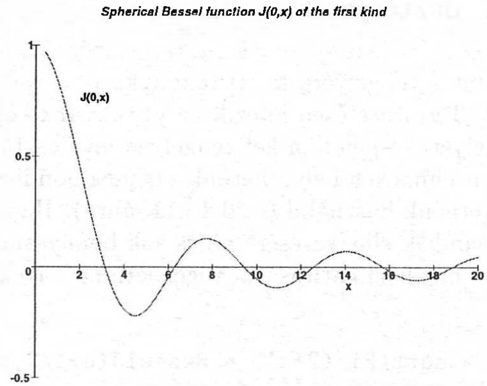
```
> J := (n,x) -> sqrt(Pi/(2*x)) * BesselJ(n+1/2,x):
> plot( J(0,x), x=0..20, ‘J(0,x)‘=-0.5..1,
>   xtickmarks=8, ytickmarks=4, title=
>   ‘Spherical Bessel function j(0,x) of the first kind‘,
>   titlefont=[HELVETICA,BOLDOBLIQUE,20] );
```



15.11. ábra: A $J_0(x)$ elsőfajú szférikus Bessel függvény

Alternatív megoldásként a `plots` csomag **textplot** eljárásával külön megrajzoltathatjuk a tengelyeket és a szöveget, majd mindezt a **display** eljárás segítségével a függvény grafikonjával közös képen jeleníthetjük meg. (Lásd a 15.12. ábrát.)

```
> curve := plot( x->J(0,x), 0..20, -0.5..1,
>   xtickmarks=8, ytickmarks=4, title=
>   ‘Spherical Bessel function J(0,x) of the first kind‘,
>   titlefont=[HELVETICA,BOLDOBLIQUE,16],
>   axesfont=[HELVETICA,BOLD,14] ):
> text := plots[textplot]( { [2,0.75,‘J(0,x)‘],
>   [14,-0.1,‘x‘] }, align={ABOVE,RIGHT},
>   font=[HELVETICA,BOLD,14] ):
> plots[display]( { curve, text } );
```



15.12. ábra: A $J_0(x)$ gráfja javított szövegelhelyezéssel

A fontok választása a különböző családoknak, stílusoknak és méreteknek megfelelő (elég kicsi) halmazból történhet.

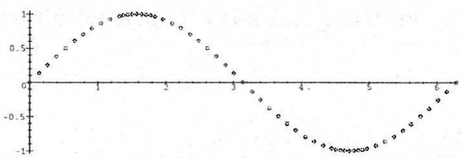
Mintavétel

A függvény grafikonjának megrajzolásakor a Maple először kiszámolja a görbe néhány pontját (a mintapontok számát a `numpoints` vagy a `sample` opcióval állíthatjuk be), majd alapföltételezés szerint egyenes szakaszokkal köti össze az így kapott pontokat. Mint a következő alfejezetben látni fogjuk, a Maple úgynevezett *adaptív ábrázolással* próbál minél simább grafikonokat produkálni. Ez azt jelenti, hogy a rendszer automatikusan növeli a mintapontok számát, ha a görbék simasága ezt megkívánja.

Rajzolási stílus

Az alapföltételezés szerinti `patch` helyett használhatjuk a `point` vagy a `line` stílust is (az utóbbi csak a sokszögek ábrázolását befolyásolja). Ha a `point` stílust választjuk, a Maple olyan kereszteket rajzol, amelyek középpontja a kiszámított mintapontokban (sokszögek esetén a sokszög csúcsaiban) van. A kereszt helyett más jelet is definiálhatunk a `symbol` opcióval. (Lásd 15.13. ábra.)

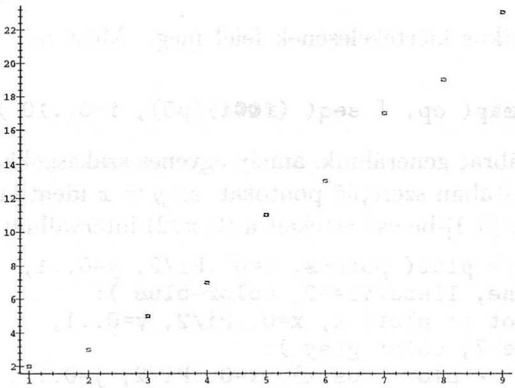
```
> plot( sin, 0..2*Pi, scaling=constrained, style=point,
>   symbol=circle );
```



15.13. ábra: A szinusz függvény pontozott grafikonja

Adatpontokat könnyen előállíthatunk a `point` stílussal. A pontpárokat listában helyezzük el, és a pontokat a `point` stílussal rajzoltatjuk ki. Az első kilenc prímszámot például a következő parancs segítségével jeleníthetjük meg (v. ö. 15.14. ábra):

```
> plotpoints := [ seq( [ i, ithprime(i) ], i=1..9 ) ]:
> plot( plotpoints, style=point, symbol=box );
```



15.14. ábra: Az első kilenc prímszámnak megfelelő adatpontok

A 15.13. alfejezetben áttekintjük azokat a beépített eljárásokat, amelyek a most látott alapvető adatvizualizációs lehetőségeket terjesztik ki.

Az alapföltételezésnek megfelelő `line` stílus szemléltetésére tekintsük a koszinusz függvény 1.2 kezdőértékkel indított 20 lépéses iterációját. Először a pontok listáját kell generálnunk:

$$\Big[\,[\,1.2, 1.2\,], [\,1.2, \cos(1.2)\,], [\,\cos(1.2), \cos(1.2)\,], [\,\cos(1.2), \cos(\cos(1.2))\,],$$

$$[\,\cos(\cos(1.2)), \cos(\cos(1.2))\,], [\,\cos(\cos(1.2)), \cos\,(\cos(\cos(1.2)))\,], \ldots\Big]$$

Az ábrázolandó pontokat a következő kezdőértékkel induló pontsorozatként generáljuk:

```
> pair0 := [[1.2, 1.2], [1.2,cos(1.2)]]; # starting pair
```

$$pair0 := [\,[\,1.2,\,1.2],\,[1.2,\,.3623577545\,]\,]$$

Az új párokat létrehozó iterációs függvény definíciója:

```
> f := pair -> [ map( cos, pair[1] ),
>                map( cos, pair[2] ) ]; # iteration
```

$$f := pair \rightarrow [\mathrm{map}(\cos,\, pair_1),\, \mathrm{map}(\cos,\, pair_2)]$$

Hatását a következő paranccsal vizsgálhatjuk,

```
> f(p0); # first new pair
```

$$[\,[\,.3623577545,\,.3623577545\,],\,[\,.3623577545,\,.9350636470\,]\,]$$

ami a

$$\left[[\cos(1.2), \cos(1.2)], [\cos(1.2), \cos(\cos(1.2))] \right]$$

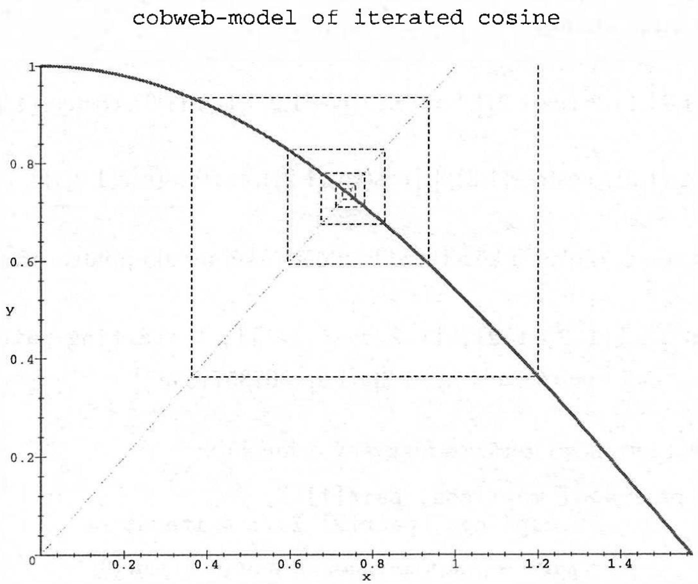
adatpontok numerikus kiértékelésének felel meg. Most már előállíthatjuk az összes pontot:

```
> points := map( op, [ seq( (f@@i)(p0), i=0..10 ) ] ):
```

Befejezésül olyan ábrát generálunk, amely egyenes szakaszokkal összekötve tartalmazza a fönti listában szereplő pontokat, az $y = x$ identikus függvényt és a koszinusz függvény $[0, 1]$-be eső értékeit a $(0, \pi/2)$ intervallum fölött.

```
> curveplot := plot( points, x=0..Pi/2, y=0..1,
>   style=line, linestyle=2, color=blue ):
> identityplot := plot( x, x=0..Pi/2, y=0..1,
>   linestyle=7, color=gray ):
> cosineplot := plot( cos(x), x=0..Pi/2, y=0..1,
>   thickness=4, color=red ):
```

Az ábrákat egyetlen képen jelenítjük meg a `plots` csomag `display` parancsával. A fönti utasításokban különböző színeket és vonalstílusokat használunk (szaggatott vonalakat), a koszinusz grafikonját pedig kivastagítva ábrázoljuk, hogy a kép komponenseit könnyebb legyen megkülönböztetni. (Lásd 15.15. ábra.)



15.15. ábra: Az iterált koszinusz „pókháló-grafikonja”

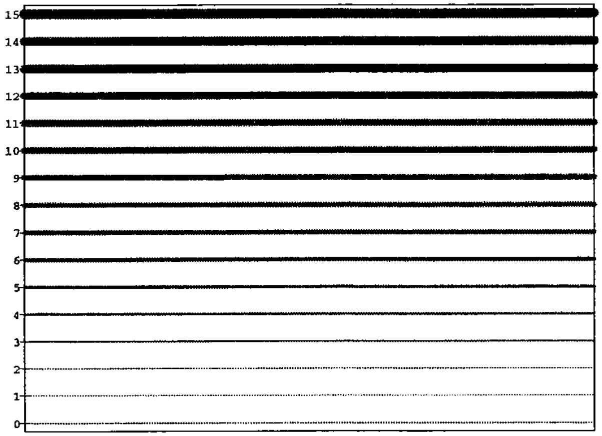
Vonalstílusok és vastagságok

Vizsgáljuk meg a rendelkezésünkre álló egyenesstílusokat és vastagságokat. Jelenleg a `linestyle` és a `thickness` opciók értéke a $[0, 7]$, illetve a $[0, 15]$ tar-

tományba eső nemnegatív egész szám lehet, ezeken kívüli értékekre moduláris aritmetikával számol a rendszer. (Lásd a 15.16. és 15.17. ábrát.)

```
> for i from 0 to 15 do
>   thick[i] := plot( i, x=0..1, thickness=i ):
> od:
> display( convert( thick, set ),
>   axes=box, tickmarks=[0,15],
>   title=`thickness option:  [0,1,..,15]` );
```

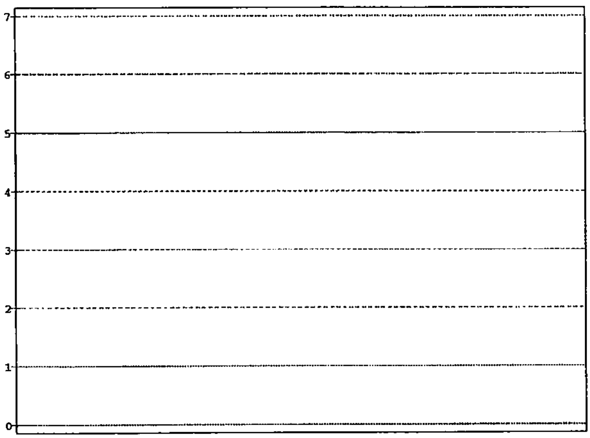
thickness option: [0,1,..,15]



15.16. ábra: Különböző vonalvastagságok

```
> for i from 0 to 7 do
>   line[i] := plot( i, x=0..1, linestyle=i ):
> od:
> display( convert( line, set ), axes=box,
>   tickmarks=[0,7], title=`linestyle option:  [0,1,..,7]` );
```

linestyle option: [0,1,..,7]

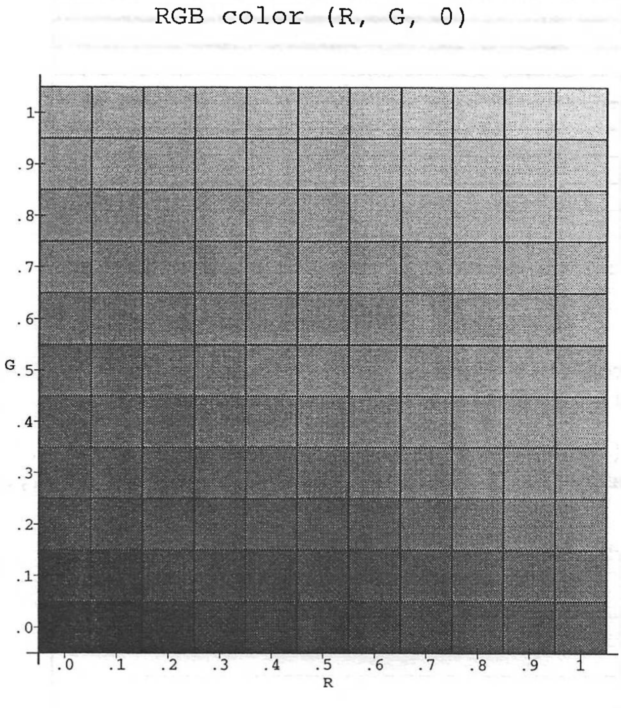


15.17. ábra: Különböző stílusú egyenesek

Színkezelés

A Maple által használt két színmodell az RGB és a HSV. Az RGB módban a szín *red* (piros), *green* (zöld) és *blue* (kék) összetevőit állítjuk be. A plots csomag **polygonplot** rutinjával könnyen készíthetünk színdiagramokat:

```
> P := seq( seq( polygonplot(
>    [ [i,j], [i+1,j], [i+1,j+1], [i,j+1] ],
>    color=COLOR(RGB,0.1*i, 0.1*j, 0 ) ),
>    i=0..10 ), j=0..10 ):
> display( {P}, scaling=constrained,
>    labels=[R,G], title=`RGB color (R, G, 0)`,
>    tickmarks = [ [seq(i+0.5=`.`.i,i=0..9),10.5=`1`],
>    [seq(j+0.5=`.`.j,j=0..9),10.5=`1`] ] );
```

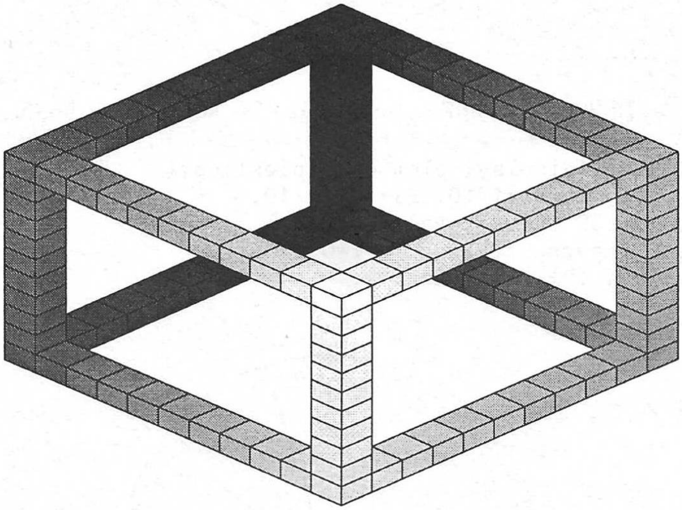


15.18. ábra: RGB színdiagram szürkeskálás vetülete

Sajnos a 15.18. ábrán csak a színeknek megfelelő szürkeségi fokozatokat láthatunk. A képernyőn vagy színes nyomtatón azonban valóban a megadott színeket kapjuk. Az RGB színmodellt három dimenzióban, egy kocka élei mentén mutatja a 15.19. ábra.

```
> e1 := seq( seq( seq( display(
>    plottools[cuboid]([i,j,k],[i+1,j+1,k+1]),
>    color=COLOR(RGB,0.1*i, 0.1*j, 0.1*k ), style=patch ),
```

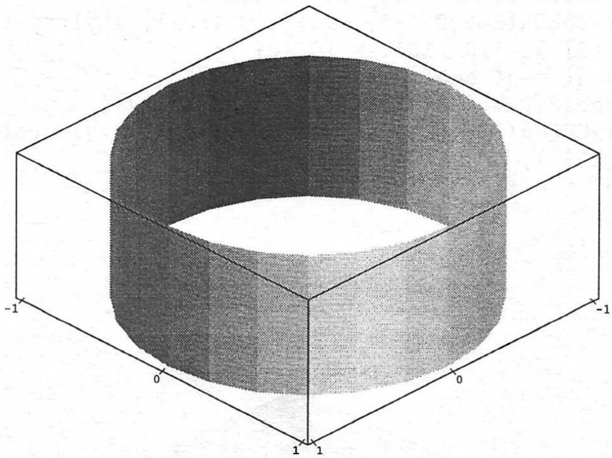
```
>   i=0..10 ), j=[0,10] ), k=[0,10] ):
> e2 := seq( seq( seq( display(
>   plottools[cuboid]([i,j,k],[i+1,j+1,k+1]),
>   color=COLOR(RGB,0.1*i, 0.1*j, 0.1*k ), style=patch ),
>   i=[0,10] ), j=0..10), k=[0,10] ):
> e3 := seq( seq( seq( display(
>   plottools[cuboid]([i,j,k],[i+1,j+1,k+1]),
>   color=COLOR(RGB,0.1*i, 0.1*j, 0.1*k ), style=patch ),
>   i=[0,10] ), j=[0,10]), k=0..10 ):
> display( { e.(1..3) } );
```



15.19. ábra: Az RGB színek ábrázolása kocka élei mentén

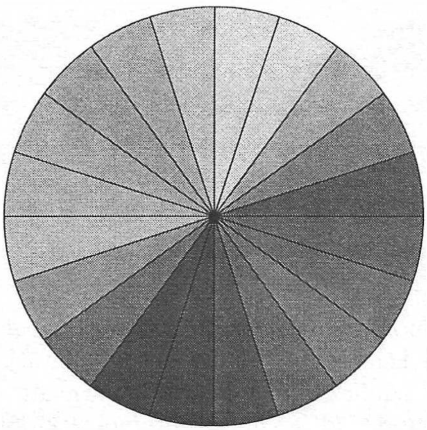
HSV módban a szín *hue*, *saturation* (telítettség) és *value* (érték) összetevőit határozzuk meg. A két utóbbi komponens azt mutatja, hogy mennyi fehéret és feketét kell a színhez adni a különböző árnyalatok eléréséhez. A HSV mód „hengeres" színteret ad, az argumentumokat cirkulárisan értelmezi. A Maple-ben `COLOR( HSV, h, 0.9, 1.0 )` helyett a rövidebb `HUE(h)` alakot is használhatjuk. A 15.20. és a 15.21. ábra a HUE színskálát kétféle módon jeleníti meg, körkörös szalagon, illetve színkörként.

```
> plot3d( 1, t=0..2*Pi, z=-2..2, coords=cylindrical,
>   style=patchnogrid, color=t/2/Pi, axes=box,
>   orientation=[45,35], tickmarks=[3,3,0] );
```



15.20. ábra: HUE színek ábrázolása körkörös szalagon

```
> P := seq( display( plottools[pieslice](
>     [0,0], 5, Pi*i/10..Pi*(i+1)/10,
>     color=COLOR(HUE,evalf(i/20)) ),
>     scaling=constrained ), i=0..20 ):
> display( {P}, axes=none );
```

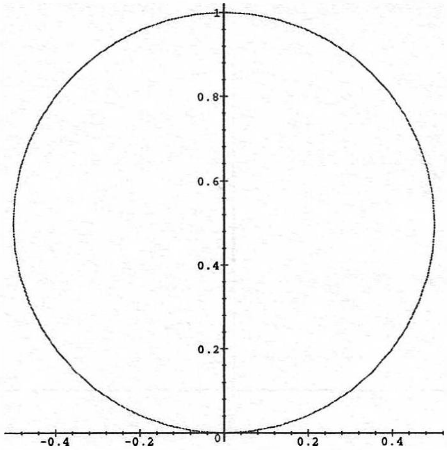


15.21. ábra: A HUE színkör

Koordinátarendszerek

Eddig a legtöbb ábrán Descartes-féle koordinátarendszert használtunk. Valójában a Maple számos rendszer közti választást tesz lehetővé. Lássunk egy polárkoordinátákkal fölírt paraméteres görbét (v. ö. 15.22. ábra):

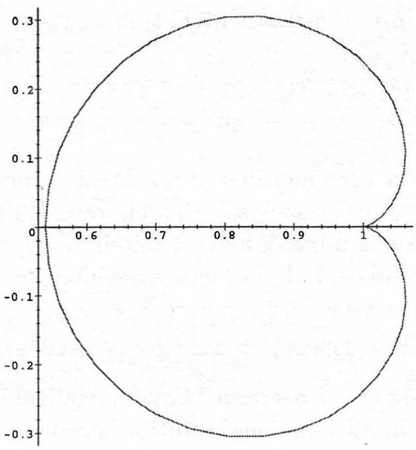
```
> plot( [ sin(t), t, t=0..2*Pi ], coords=polar,
> scaling=constrained );
```



15.22. ábra: Polárkoordinátákkal definiált paraméteres görbe

Hasonlítsuk össze a 15.22. ábrán látható grafikont a 15.1. alfejezet utolsó példájával, melyben Descartes-féle koordinátákat használtunk. A `plots` csomag **changecoords** rutinjával meglévő plotstruktúrákat változtathatunk meg úgy, hogy az eredeti (Descartes-félének tekintett) helyett más koordinátarendszert adunk meg. (Lásd 15.23. ábra.)

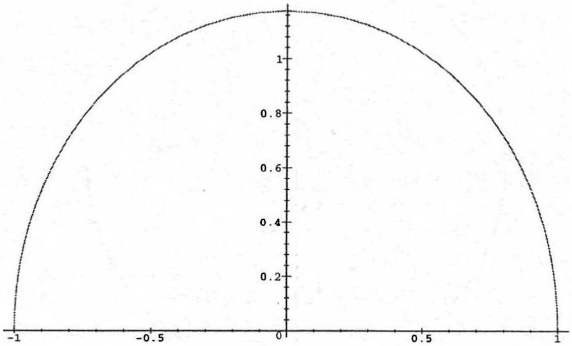
```
> P := ": # give the previous plot a name
> changecoords( P, elliptic );
```



15.23. ábra: A görbe rajza a koordinátacsere után

Figyeljük meg, hogy ez mást jelent, mint az alábbi parancs!

```
> plot( [ sin(t), t, t=0..2*Pi ], coords=elliptic,
>    scaling=constrained );
```



15.24. ábra: A görbe képe elliptikus koordinátákban

A **changecoords** úgy tekinti, hogy az eredeti plotstruktúrát az alapföltételezés szerinti (általában a Descartes-féle) koordinátarendszerben hoztuk létre, s ezt a kétdimenziós adatstruktúrát alakítja át az új koordinátarendszernek megfelelően. A **changecoords** tehát nem veszi figyelembe, ha a kétdimenziós plotstruktúrát eredetileg esetleg más koordinátarendszerben generáltuk. (V. ö. 15.24. ábra.)

Alapbeállítások

Ha valamelyik plot opció alapföltételezés szerinti értékére vagyunk kíváncsiak, a `plots` csomag **setoptions** utasítását hívhatjuk segítségül.

```
> restart; with(plots):
> setoptions( numpoints );
```

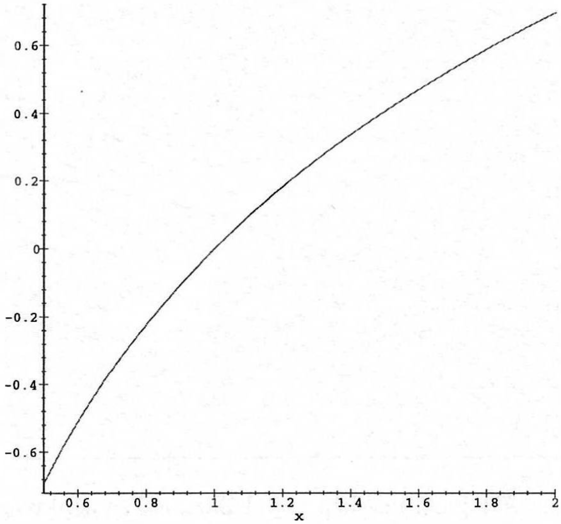
49

Elsődlegesen azonban a kétdimenziós ábrázolással kapcsolatos opciók alapbeállításának megváltoztatására szolgál ez az eljárás. Ha például a kétdimenziós ábrákon a tengelyeket a `normal` helyett mindig a `frames`-nek megfelelően akarjuk föltüntetni, továbbá mindig azonos vízszintes és függőleges skálázást kívánunk alkalmazni, írjuk be a

```
> setoptions( axes = frame, scaling = constrained ):
```

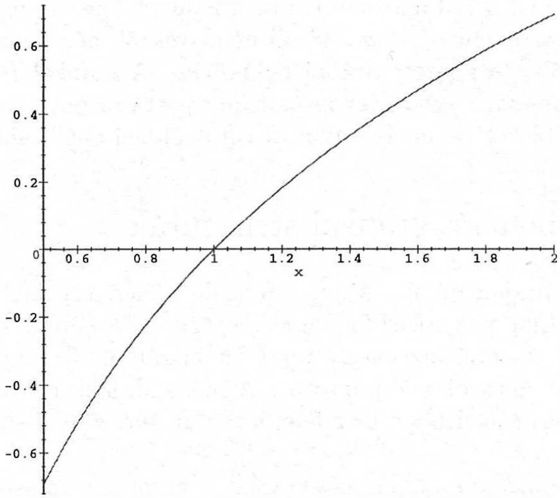
parancsot, vagy helyezzük el ezt a sort a Maple inicializáló fájlunkban. Mint a következő ábrasorozat is mutatja, a megjelenítést végző parancsokban megadott opciókkal bármikor fölülbírálhatjuk az alapbeállításokat. (V. ö. 15.25., 15.26. és 15.27. ábra.)

```
> plot( ln(x), x = 1/2..2 );
```



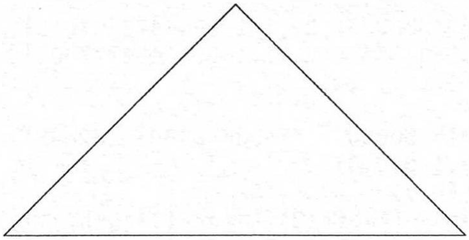
15.25. ábra: A természetes alapú logaritmus képe az $(\frac{1}{2}, 2)$ intervallumon

```
> # replot with normal axes
> display( ", axes=normal );
```



15.26. ábra: A logaritmus képe más tengelystílussal

```
> # replot with new display area and equal scaling
> display( "", view=[0..2,-2..2], scaling=unconstrained );
```



15.28. ábra: A grafikai primitívekből fölépített háromszög

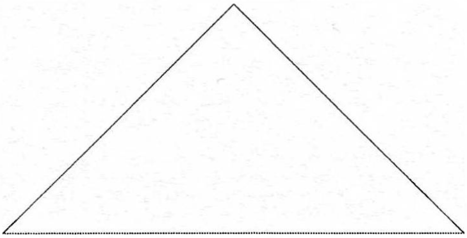
A CURVES(...), AXESSTYLE és SCALING argumentumok azon grafikus objektum argumentumainak részei, amelyet a következő paranccsal hozhatunk létre:

```
> P := plot( [ [1,1], [2,2], [3,1], [1,1] ],
>   axes=none, scaling=constrained ):
> lprint( P ); # print the plot data structure

PLOT(CURVES([[1., 1.], [2., 2.], [3., 1.], [1., 1.]],
     COLOUR(RGB,1.0,0,0)), SCALING(CONSTRAINED),
     AXESSTYLE(NONE))
```

A 15.29. ábrát ennek a grafikus objektumnak kiértékelésével kaptuk.

```
> P; # plot the graphics object
```



15.29. ábra: A **plot** segítségével kirajzolt háromszög

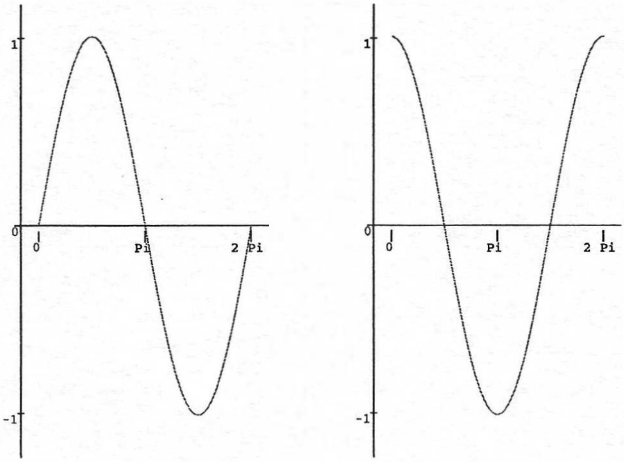
Általánosabb példát kapunk a képek mögötti grafikus objektumokra, ha az

$$x \longmapsto \sqrt{2} - \sqrt{2\sqrt{x}}$$

egyváltozós valós függvénynek a 15.30. ábrán látható grafikonjához tartozó plot-struktúrát tekintjük:

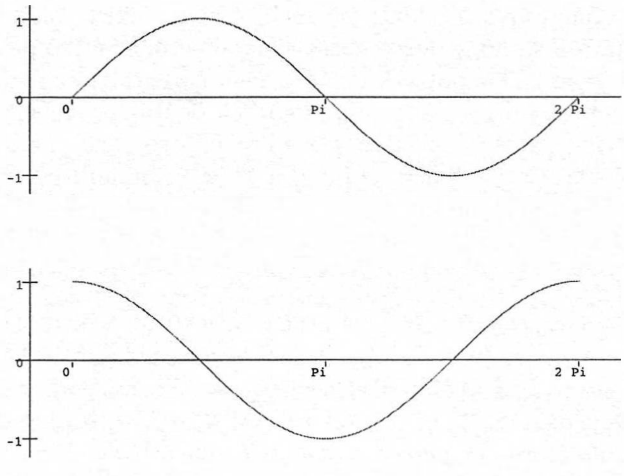
```
> f := x -> sqrt(2) - sqrt(2*sqrt(x)):
> P := plot( f(x), x=0..1, y=0..sqrt(2),
>   title=`graph of sqrt(2) - sqrt(2*sqrt(x))` ):
> lprint( P ); # print plot data structure
```

```
> ticks := [ [0=‘0‘, 3.14=‘Pi‘, 6.28=‘2 Pi‘],
>            [-0.99=‘-1‘,0=‘0‘,0.99=‘1‘] ]:
> sine := plot( sin(x), x=0..2*Pi, tickmarks=ticks ):
> cosine := plot( cos(x), x=0..2*Pi, tickmarks=ticks ):
> display( array([sine,cosine]) ); # row of plots
```



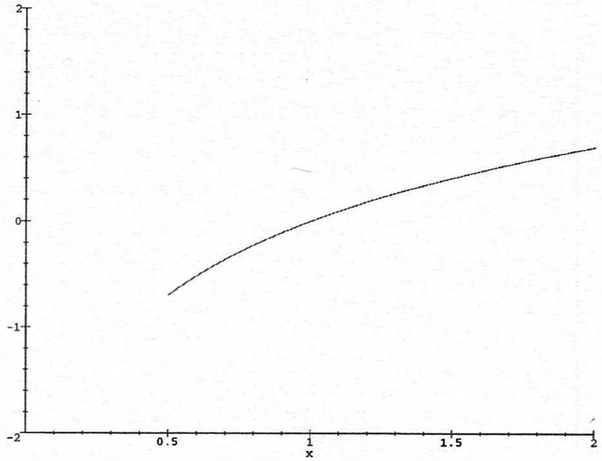
15.46. ábra: Grafikonokból álló „sorvektor”

```
> display( array(1..2,1..1,[[sine],[cosine]]) );
> # column of plots
```



15.47. ábra: Grafikonokból álló „oszlopvektor”

A koordinátatengelyek beosztását azért kellett ilyen furcsán megadni, hogy a grafikus tömbök megjelenítésekor helyes jeleket kapjunk.



15.27. ábra: A logaritmus képe új skálázással és képkivágással

Jegyezzük meg, hogy a **display** parancsban *kizárólag* azokat az opciókat bírálhatjuk fölül az előző módon, amelyek megváltozása nem kívánja meg a grafikus objektum újraszámolását. Például nem várhatjuk el azt, hogy a `numpoints` opcióval a grafikus objektum újraszámolása nélkül meg lehessen változtatni a mintapontok számát.

A kétdimenziós ábrázolás rutinjaihoz tartozó sok opció leírását megtalálhatjuk az online help-ben a **?plot,options** utasítással. A fejezet végén fölsoroljuk az összes elérhető plot opciót. Ezek közül némelyekről még nem esett szó, de használni fogjuk őket a fejezet további példáiban. A munkalapos felületen a `linestyle`, `thickness`, `symbol`, `axes` és a `projection` opció interaktívan is megadható. A többieket valamelyik plot utasítással kell beállítani.

15.3. Kétdimenziós grafikai struktúrák

Ahhoz, hogy meg tudjuk ítélni a Maple ábrázolási lehetőségeinek megbízhatóságát, elengedhetetlen a rajzolási folyamat megértése. Az ábrák elkészítése két fázisban történik. Az első fázisban a rendszer kiszámolja és egy `PLOT` objektumba helyezi el az ábrázolandó pontokat. A második fázis ezt az objektumot jeleníti meg a képernyőn. Ebben az alfejezetben az ábrázolás első fázisára koncentrálunk.

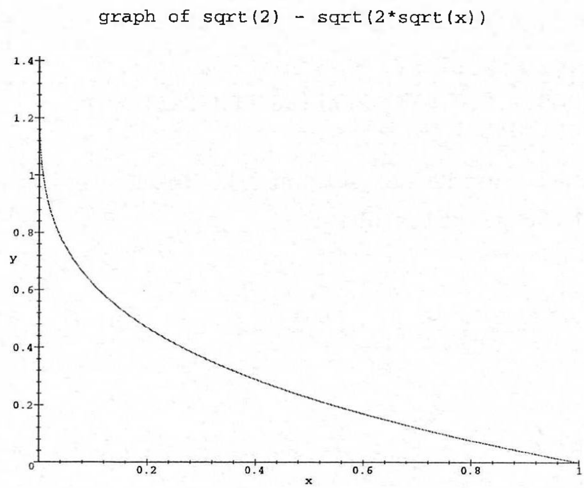
A Maple kétdimenziós grafikus objektumai a `PLOT` függvény olyan hívásai, amelyekben az argumentumok a tengelyeket, az ábrázolandó függvényt, a kiszámított pontokat, az ábrázolás stílusát stb. írják le. A következő objektum az $(1,1)$, $(2,2)$ és $(3,1)$ koordinátájú csúcsok által meghatározott háromszöget írja le. (Lásd a 15.28. ábrát.)

```
> PLOT( CURVES( [ [1,1], [2,2], [3,1], [1,1] ] ),
>    AXESSTYLE(NONE), SCALING(CONSTRAINED) );
```

```
PLOT(CURVES([[0, 1.414213562373095],
    [.6811610677083333e-3, 1.185744472197018],
    [.1362322135416667e-2, 1.142516494777506],
                 ..............
    [.9782721018750000, .7745369848915162e-2], [1., 0]],
    COLOUR(RGB,1.0,0,0)),
    AXESLABELS(x,y),
    TITLE(`graph of sqrt(2) - sqrt(2*sqrt(x))`),
    VIEW(0 .. 1.,0 .. 1.414213562))
```

A Maple az adott intervallumon értelmezett egyváltozós valós függvényekhez egy grafikus objektumot rendel. Az objektum szabályos Maple adatstruktúra, a rendszer ezt használja az ábra kiszámolt pontjai közötti lineáris interpolációra.

```
> P; # display the graph
```



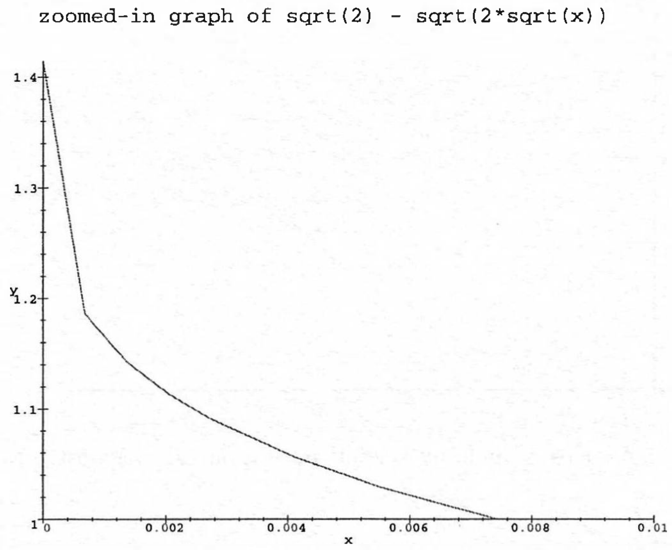
15.30. ábra: A $2^{1/2}(1 - x^{1/4})$ grafikonja

Először a **plot** hívásának a függvényt leíró argumentuma értékelődik ki. Ezután a mintapontokban fölvett függvényértékek numerikus közelítéseit számolja ki a rendszer. A megjelenítés fölgyorsítása érdekében rendszerint a hardveres lebegőpontos aritmetikát használja.

A Maple úgy tud szép sima görbéket előállítani, hogy szükség esetén bizonyos algoritmus szerint növeli a mintapontok számát. Először az adott intervallumon ekvidisztánsan kijelölt 49 mintapontban számolja ki a függvényértékeket. Ezeket a pontokat egyenes szakaszokkal köti össze. Ha két egymáshoz csatlakozó szomszédos szakasz által bezárt szög túl nagy, a rendszer további mintapontokat vesz föl az illető pont körül. Természetesen a fölosztás nem folytatódhat minden határon túl; maximális mértéke a `resolution` opcióval szabályozható, melynek alapértelmezése 200. Így tehát alapértelmezés szerint a mintapontok száma 49 és 200 közé eshet.

Az ábrák „zoomolását" (kinagyítását vagy lekicsinyítését) a `plots` csomag **display** parancsával érhetjük el:

```
> display( P, view=[0..0.01, 1..sqrt(2)], title=
>   'zoomed-in graph of sqrt(2) - sqrt(2*sqrt(x))' );
```



15.31. ábra: A $2^{1/2}(1 - x^{1/4})$ kinagyított grafikonja

Az előző példában x kicsi értékeire már az adaptív mintavételezési sémát alkalmazta a Maple. Erről tájékoztat is bennünket, ha az `infolevel[plot]` változó értékét megnöveljük:

```
> infolevel[plot] := 2:
> plot( f(x), x=0..1, y=0..sqrt(2) ):

plot/adaptive:   evalhf succeeded
plot/adaptive:   produced    59.    output segments
plot/adaptive:   using    59.   function evaluations
```

Összehasonlításul állítsuk át az `adaptive` opciót a `false` értékre:

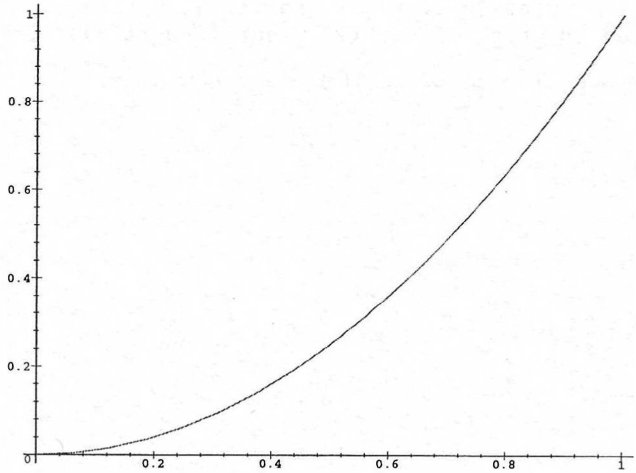
```
> plot( f(x), x=0..1, y=0..sqrt(2), adaptive=false ):

plot/adaptive:   evalhf succeeded
plot/adaptive:   produced    49.    output segments
plot/adaptive:   using    49.   function evaluations
```

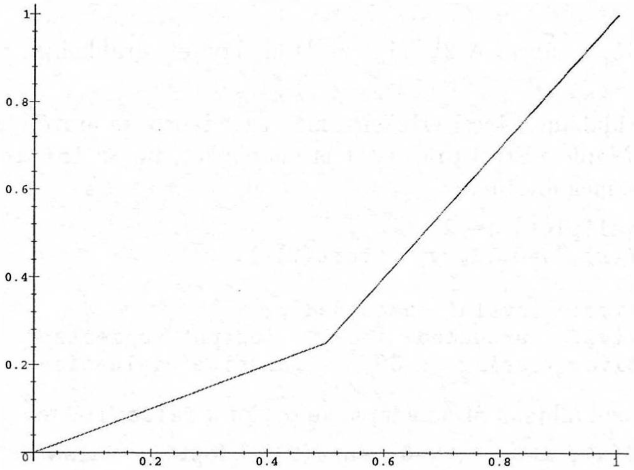
Most az alapértelmezésnek megfelelően csak 49 ponttal számolt a Maple. Az adaptív mintavétel hatása még nyilvánvalóbb, ha a `sample` plot opcióval explicit módon megadjuk a mintapontok kiinduló halmazát:

```
> infolevel[plot]:=1:  # reset information level
> plot( x^2, x=0..1, sample = [0,1/2,1], adaptive=true );
> plot( x^2, x=0..1, sample = [0,1/2,1], adaptive=false );
```

Hasonlítsuk össze a két esetnek megfelelő 15.32. és 15.33. ábrát.



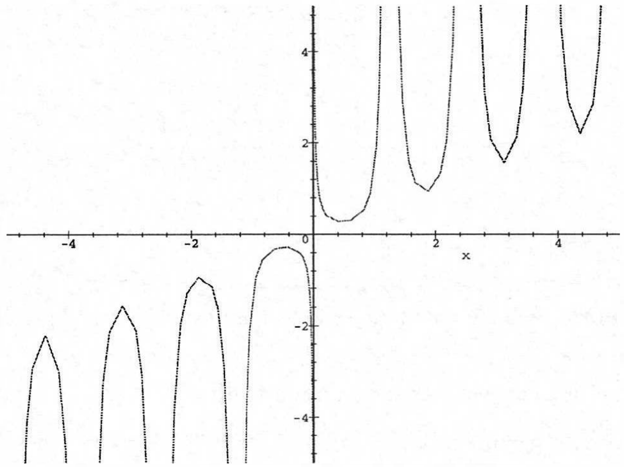
15.32. ábra: Az $x \mapsto x^2$ néhány kezdeti mintaponttal megadott grafikonja



15.33. ábra: Az $x \mapsto x^2$ néhány kezdeti mintaponttal megadott nem adaptív grafikonja

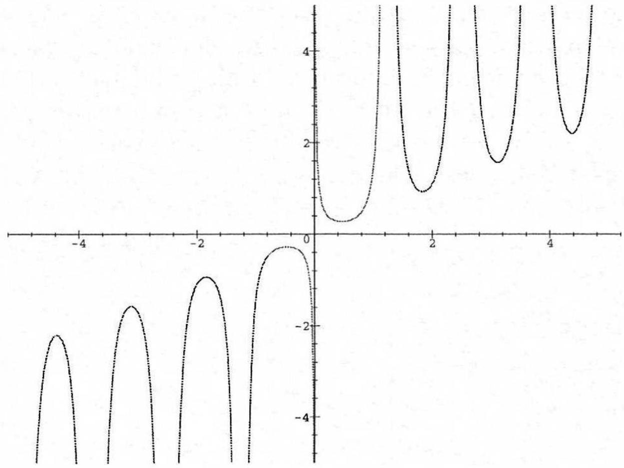
Valahányszor a 15.34. ábrához hasonló szabálytalanságokat veszünk észre egy grafikonon (ez egyébként az $\dfrac{x}{1-\cos 5x}$-et ábrázolja a $(-5,5)$ intervallumon), megpróbálhatunk a mintapontok számának növelésével jobb ábrát és szebb, simább görbét elérni. Ehhez csak a `numpoints` opciót kell nagyobb értékre állítani. (V. ö. 15.35. ábra).

```
> plot( x/(1-cos(5*x)), x=-5..5, -5..5 );
```



15.34. ábra: Az $x \mapsto \frac{x}{1-\cos 5x}$ vázlatos grafikonja

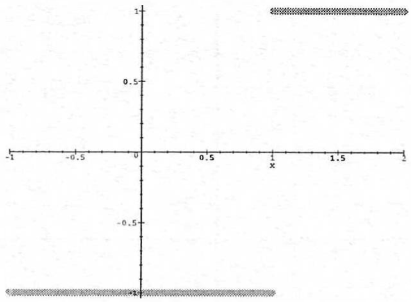
```
> plot( x/(1-cos(5*x)), x=-5..5, -5..5, numpoints=200 );
```



15.35. ábra: Az $x \mapsto \frac{x}{1-\cos 5x}$ „kisimított" grafikonja

A plotstruktúrák hasonlóak a Maple többi adatstruktúrájához. Ezek is átalakíthatók, fájlba menthetők és onnan visszaolvashatók. Példaként tekintsük a következő lépcsősfüggvényt:

```
> step := plot( 2*Heaviside(x-1) - 1, x=-1..2,
> discont=true, thickness=15 ):
> step; # display the graph
```



15.36. ábra: A 2 Heaviside($x - 1$) − 1 lépcsősfüggvény grafikonja

Most mentsük el a step-et a graph.m nevű fájlba.

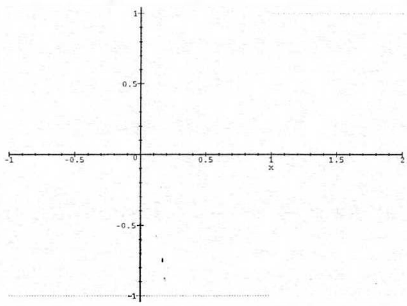
```
> save step, 'graph.m':
```

A step változóhoz a NULL értéket rendeljük, így a graph.m fájl beolvasása után látható lesz, hogy ismét régi értékét vette föl:

```
> step := NULL: # reassign the variable
> read 'graph.m': # load the graph
> step; # display the graph
```

Könnyen meggyőződhetünk róla, hogy ismét a 15.36. ábrát kaptuk. Az adatstruktúra egyszerű átalakítását jelenti a thickness opció megváltoztatása. Mivel a görbe vastagságát megadó információ a plot adatstruktúra CURVES(...) komponensének belsejében található, továbbá a **plots[display]** parancs ezt a komponenst nem változtatja meg, a vonalvastagság gyors megváltoztatásához vagy helyettesítést kell végeznünk, vagy újra kell generálnunk a plotstruktúrát a **plot** paranccsal. (V. ö. 15.37. ábra.)

```
> subs( THICKNESS(15)=THICKNESS(1), step );
```



15.37. ábra: A 2 Heaviside($x - 1$) − 1 lépcsősfüggvény újrarajzolása más vonalvastagsággal

15.4. A plottols csomag

Folytassuk tárgyalásunkat a kétdimenziós grafikus objektumok leírásával. Az előző alfejezetben találkoztunk a CURVES(...) alacsony szintű grafikai primitívvel, ott a

$$\mathtt{CURVES}(\,[\,[1.,1.],\,[2.,2.],\,[3.,1.],\,[1.,1.]\,],\,\mathtt{COLOUR}(\mathtt{RGB},1.0,0,0)\,)$$

példát említettük. A teljes plotstruktúrához hasonlóan a grafikus primitíveket is függvényhívásokként implementálták a Maple-ben. Ezek általános alakja

$$\textit{ObjectName}(\ \textit{ObjectInformation},\ \textit{LocalInformation}\ ).$$

A fönti példában CURVES az objektum neve, a pontok listája jelenti az objektumhoz tartozó információt és a szín megadása olyan lokális információ, amely a görbe színét határozza meg. A kétdimenziós megjelenítésnél előforduló egyéb objektumnevek POINT, POLYGONS és TEXT. A lokális információt is függvényhívásokként implementálták. Az itt szereplő tipikus nevek többek között AXESSTYLE, AXESTICKS, COLOUR, FONT, THICKNESS és VIEW. A lokális információra vonatkozó összes nevet a **?plot,structure** parancs írja ki. Ezeket a grafikus primitíveket mindig nagybetűvel írjuk; használhatjuk az amerikai helyesírásnak megfelelő alakjukat is. Segítségükkel grafikus objektumokat építhetünk föl. Például hozzunk létre egy határvonalak nélküli kék négyzetet mindkét irányban azonos skálázással a koordinátatengelyek föltüntetése nélkül:

```
>  PLOT( POLYGONS( [[0,0], [1,0], [1,1], [0,1]],
>        COLOR( RGB,0,0,1) ),
>       AXESSTYLE(NONE), STYLE(PATCHNOGRID),
>       SCALING(CONSTRAINED) );
```



15.38. ábra: A fölhasználó által megadott színű, tengelyek és határvonalak nélküli négyzet

Másik példánk olyan egyenlőoldalú háromszög, amelynek oldalegyeneseit vörös színű vastag szaggatott vonalakkal rajzoltuk meg, belsejében pedig kék színű, 24 pontos, dőlt Helvetica betűtípussal az „equilateral triangle" szöveget helyeztük el.

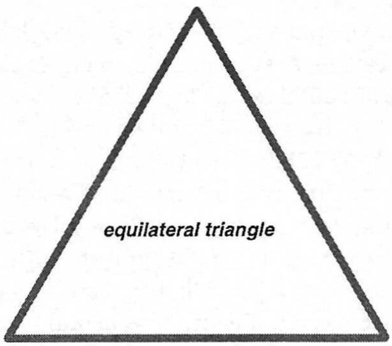
```
>  PLOT(
>    CURVES( [[0,0],[1,0],[1/2,1/2*sqrt(3)],[0,0]],
>      COLOR( RGB,1,0,0), THICKNESS(15), LINESTYLE(3) ),
>    TEXT( [1/2,1/6*sqrt(3)], `equilateral triangle`,
>      COLOR(RGB,0,0,1), FONT([HELVETICA,BOLDOBLIQUE,24]) ),
```

```
>   AXESSTYLE(NONE), SCALING(CONSTRAINED) );
```

```
Plotting error, non-numeric vertex definition
```

Szándékosan adtuk meg hibásan az utasítást: a grafikai primitívek csak numerikus adatokat tartalmazhatnak, ezért $\frac{1}{2}\sqrt{3}$ helyett például a 0.866 lebegőpontos közelítését kell megadnunk. A helyes input tehát a következő:

```
> PLOT(
>   CURVES( [[0,0],[1,0],[1/2,0.866],[0,0]],
>     COLOR(RGB,1,0,0), THICKNESS(15), LINESTYLE(3) ),
>   TEXT( [1/2,0.289], 'equilateral triangle',
>     COLOR(RGB,0,0,1), FONT(HELVETICA,BOLDOBLIQUE,24) ),
>   AXESSTYLE(NONE), SCALING(CONSTRAINED) );
```



15.39. ábra: Egyenlőoldalú háromszög közepébe írt szöveggel

Ezek a példák is azt illusztrálják, hogy mennyire fárasztó lenne, ha grafikus objektumainkat állandóan az alacsony szintű grafikus primitívekből kellene fölépítenünk. Nagyon kényelmes lenne egy olyan gyűjtemény, amely a gyakran használt építőelemeket tartalmazná. Van is ilyen a Maple-ben: a `plottols` csomag. A `plots` csomaggal együtt alkalmazva bonyolult grafikai feladatokat is meglepően könnyen megoldhatunk segítségével. Például az előző ábrákat így is előállíthattuk volna (ugyanazokat a képeket kapjuk, ezért a rajzokat kihagytuk):

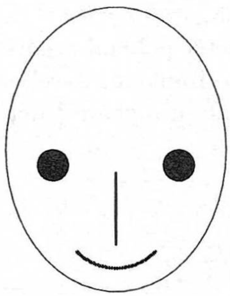
```
> with(plots): with(plottools): # loadpackages
> # blue square
> display( rectangle( [0,0], [1,1], color=blue),
>   axes=none, style=patchnogrid,scaling=constrained );
```

```
> # equilateral triangle with text inside
> lines :=
>   curve([[0,0],[1,0],[1/2,1/2*sqrt(3)],[0,0]],
>   color=red, thickness=15, linestyle=3 ):
```

```
> text :=
>   textplot( [1/2,1/6*sqrt(3), 'equilateral triangle'],
>   color=blue, font=[HELVETICA,BOLDOBLIQUE,24]):
> display( {lines, text}, axes=none, scaling=constrained );
```

Ha az előző példák nem győzték meg az Olvasót a plottols csomag hasznosságáról, próbálja meg a grafikai primitívekkel fölrajzolni a 15.40. ábrán látható nevető arcot:

```
> head := ellipse( [0,0.5], 0.7, 0.9, color=black ):
> eyes := disk( [0.4,0.4],0.1, color=blue ),
>         disk( [-0.4,0.4],0.1, color=blue ):
> mouth := arc( [0,0.1], 0.35, 5/4*Pi..7/4*Pi,
>         color=red, thickness=7 ):
> nose := line( [0,0.35], [0,-0.1], color=black,
>   thickness=5 ):
> display( {head, eyes, nose, mouth},
>   scaling=constrained, axes=none ); # happy face
```



15.40. ábra: Nevető arc

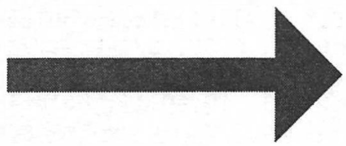
A plottools csomagban vannak olyan függvények is, amelyekkel egy meglévő grafikus objektumot alakíthatunk át (a 15.1. táblázatban soroltuk föl őket). A plots csomag **changecoords** eljárásával együtt ezek a Maple eljárások alkalmazhatók grafikus adatstruktúrák közti leképezések realizálására.

Eljárás	**Hatása**
rotate	elforgatás az óramutató járásával szemben
scale	skálázás
stellate	csillagpoligonok generálása
transform	a plotstruktúrát transzformáló függvény generálása
translate	a grafikus struktúra transzlációja

15.1. táblázat: A plottools csomag grafikus objektumokat kezelő függvényei

Készítsünk ezekkel az eljárásokkal különböző színű nyilakból álló olyan sorozatot, ahol az egyes nyilak eltérő méretűek, de úgy vannak eltolva, hogy hegyük vége mindig az egységkörön legyen. A nyilakból álló spektrum a 15.43. ábrán látható. Kiindulásul egy vízszintes állású vastag nyilat rajzolunk. (Lásd a 15.41. ábrát.)

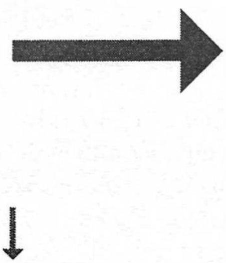
```
> a[0] := display( arrow( [0,0], [1,0], 0.1, 0.4, 0.2 ),
>   color=COLOR(HUE,0), axes=none, scaling=constrained ):
> a[0]; # display the horizontal arrow
```



15.41. ábra: Vízszintes nyíl

Változtassuk meg a nyíl helyzetét például a következő módon: forgassuk el az óramutató járásával megegyező irányban $\frac{\pi}{2}$-vel, skálázzuk minden irányban $\frac{1}{4}$-szeresére, végül toljuk el negatív irányban $\frac{3}{4}$-del a függőleges tengely mentén. (Lásd a 15.42. ábrát.)

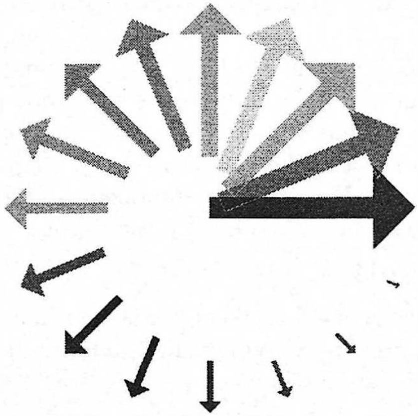
```
> display( a[0],
>   translate( scale(
>     rotate( a[0],-Pi/2 ),
>   1/4, 1/4 ), 0,-3/4 ) );
```



15.42. ábra: Az eredeti és az elforgatott, skálázott, majd eltolt nyíl

Mivel a **rotate**, **scale**, **translate** és a **display** eljárások hívása nem tartalmaz olyan opciót, amellyel a nyíl színét meg tudnánk adni, a nyílspektrum alábbi generálásakor helyettesítéssel specifikáljuk a megfelelő színeket:

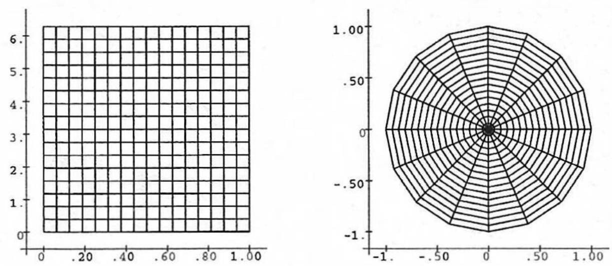
```
> for i to 15 do
>   a[i] := subs( COLOR(HUE,0) = COLOR(HUE,i/16),
>   translate( scale(
>     rotate( a[0], i*Pi/8 ),
>   (16-i)/16, (16-i)/16 ),
>   i/16*cos(Pi*i/8), i/16*sin(Pi*i/8) ) )
> od:
> display( [ seq(a[i],i=0..15) ], scaling=constrained );
```



15.43. ábra: Különböző HUE színekből álló nyílspektrum

A **transform** eljárással koordinátarendszerek közti transzformációkat végezhetünk. Példaként tekintsük egy derékszögű rácshálózat átalakítását a polárkoordinátáknak megfelelő beosztássá.

```
> hpoints := [ seq( [ seq( [i/16,j*Pi/8],
>           i=0..16) ], j=0..16 )]:
> hlines := map( curve, hpoints ):
> vpoints := [ seq( [ seq( [i/16,j*Pi/8],
>           j=0..16) ], i=0..16 )]:
> vlines := map( curve, vpoints ):
> grid := display( hlines,vlines, axes=frame ):
> f := transform( (r,phi) -> [ r*cos(phi), r*sin(phi) ] ):
> display( array( [ grid, f(grid) ] ),
>   scaling=constrained );
```



15.44. ábra: Négyzethálós beosztás transzformálása Descartes-félékről polárkoordinátákra

Itt a `plots` csomag **display** eljárásának azt a tulajdonságát használtuk föl, hogy képes több ábrából álló *grafikus tömbök* megjelenítésére is. A jobboldali beosztás szebben is kirajzoltatható a `plots` csomag **coordplot** rutinjával.

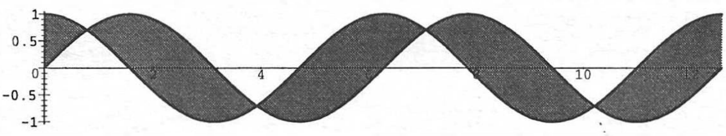
15.5. Speciális kétdimenziós ábrák

Az előző példák már jól illusztrálták, hogy a grafikus megjelenítés az egyváltozós függvények grafikonjának fölrajzolásánál sokkal több lehetőséget kínál. Ebben a részben a `plots` csomag részeként elérhető további kétdimenziós grafikai lehetőségeket sorolunk föl. Emlékeztetünk rá, hogy a fejezet többi részéhez hasonlóan most is föltesszük, hogy minden mintaként közölt Maple-szekció elején a **with(plots)** paranccsal betöltöttük ezt a csomagot.

Ábrák összekapcsolása

Az ábrák összekapcsolásának példájaként közös koordinátarendszerben ábrázoljuk a szinusz és a koszinusz függvényt úgy, hogy a hátteret fehérre, a két grafikon közti területet pedig vörösre festjük:

```
> sine := plot( sin , 0..4*Pi, color=black, thickness=3 ):
> s := plot( sin, 0..4*Pi, filled=true, color=red ):
> cosine := plot( cos, 0..4*Pi, color=black,
>   thickness=3 ):
> c := plot( cos, 0..4*Pi, filled=true, color=red):
> f := x -> if cos(x)>0 and sin(x)>0 then
>         min(cos(x),sin(x))
>       elif cos(x)<0 and sin(x)<0 then
>         max(cos(x),sin(x))
>       else 0
>       fi:
> b := plot( f, 0..4*Pi, filled=true, color=white ):
> display( [ sine, cosine, b, s, c ],
>   scaling = constrained );
```



15.45. ábra: Ábrák összekapcsolása

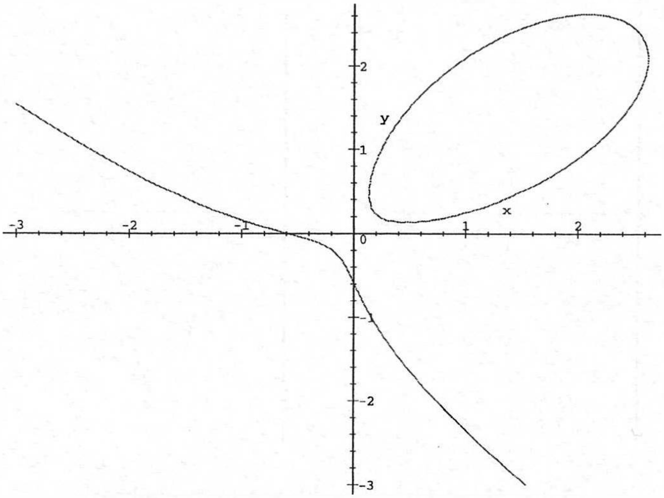
Az előző **display** parancsban lényeges a grafikonok sorrendje: megjelenítésüket a listában balról jobbra haladva, az ábrákat mindig az előzőek mögé helyezve végzi a rendszer. Ezért a `b` háttérszínt a szinusz függvény és a vízszintes tengely közti árnyékolás tetejére festve látjuk. Erről úgy is meggyőződhetünk, hogy a fönti konstrukcióban három színt használunk árnyékolásra.

A **display** eljárással „derékszögű grafikus tömbként” egy rajzon egymás mellett több ábrát is megjeleníthetünk. (V. ö. 15.46 és 15.47. ábra.) Tekintsük a következő két példát:

Algebrai síkgörbék ábrázolása

A `plots` csomagban találjuk az **implicitplot** eljárást, amely egyenlettel megadott kétdimenziós síkgörbék megrajzolására szolgál. (Lásd 15.48. ábra.)

```
> implicitplot( x^3 + y^3 - 5*x*y + 1/5 = 0,
>   x=-3..3, y=-3..3, grid=[50,50] );
```



15.48. ábra: Az $x^3 + y^3 - 5xy + 1/5 = 0$ implicit görbe képe

A Maple lényegében a következő módszert alkalmazza: az egyenletet háromdimenziós térbeli függvénynek tekinti és ennek veszi a z-síkkal való metszetét. Az eljárás egyik hátránya, hogy durva ábrát generál (v. ö. az előző példa ábráját a `grid` opció elhagyásával kapottal), továbbá nem garantálja az ábra helyességét a szinguláris pontok környezetében és a görbék önátmetszésénél. Algebrailag ez azt jelenti, hogy a Maple-nek az $f(x, y) = 0$ görbe azon (x, y) pontjainak ábrázolásával lehetnek gondjai, ahol mind a $\frac{\partial f}{\partial x}(x, y)$, mind a $\frac{\partial f}{\partial y}(x, y)$ parciális derivált nulla. Ilyen például a

$$2x^4 + y^4 - 3x^2y - 2y^3 + y^2 = 0$$

függvény, amelynek két szinguláris pontja van, a $(0, 0)$ és a $(0, 1)$, itt a görbe ágai átmetszik egymást (v. ö. [195]). A Maple még akkor sem képes helyesen fölrajzolni az ábrát az **implicitplot** segítségével, ha nagyon finom beosztást használunk. Így például 300×300-as ráccsal a SPARCStation 20 Model 71 típusú munkaállomáson 18 perces számolás és 20 Mb memória fölhasználása után sem kaptunk kielégítő pontosságú eredményt az origó körül (az ábrát elhagytuk). Ebben az esetben sokkal célszerűbb a görbe polárkoordinátás alakját kiszámolni és ezt ábrázolni a **polarplot** paranccsal. (Lásd 15.49. ábra.)

```
> subs( x=r*cos(phi), y=r*sin(phi),
>   2*x^4 + y^4 - 3*x^2*y - 2*y^3 + y^2 ):
```

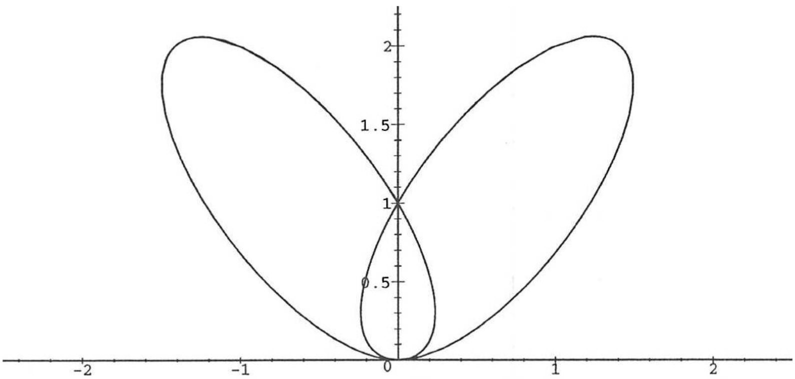
```
> factor(");
```

$$r^2\,(2\,r^2\cos(\phi)^4+r^2\sin(\phi)^4-3\,r\cos(\phi)^2\sin(\phi)-2\,r\sin(\phi)^3+\sin(\phi)^2)$$

```
> eqn := op(2,"):
> sols := map( combine, { solve(eqn,r) } );
```

$$sols := \{\frac{9\sin(\phi)+\sin(3\,\phi)-\sin(\phi)\sqrt{30+8\cos(2\,\phi)-22\cos(4\,\phi)}}{9+3\cos(4\,\phi)+4\cos(2\,\phi)},$$
$$\frac{9\sin(\phi)+\sin(3\,\phi)+\sin(\phi)\sqrt{30+8\cos(2\,\phi)-22\cos(4\,\phi)}}{9+3\cos(4\,\phi)+4\cos(2\,\phi)}\}$$

```
> sols := map( unapply, sols, phi ):
> polarplot( sols, 0..2*Pi, view=[-5/2..5/2,0..9/4],
>   scaling=constrained, color=black );
```

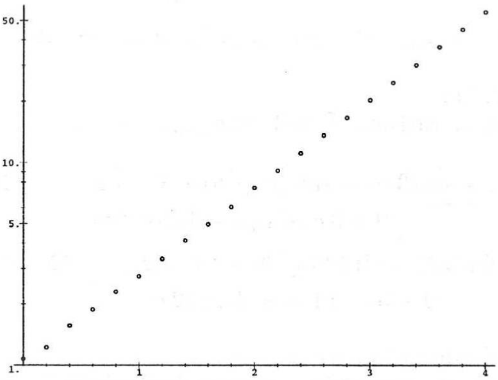


15.49. ábra: Polárkoordinátákkal számított paraméteres ábrázolás

Logaritmikus ábrázolás

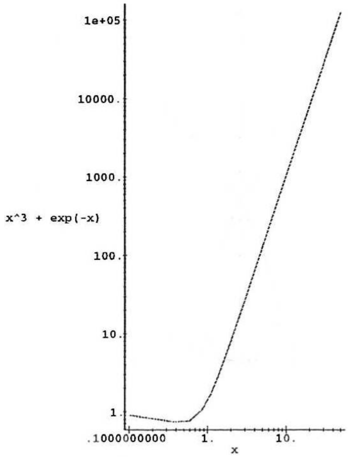
A Maple plots csomagjának **logplot**, **semilogplot** és **loglogplot** eljárásaival logaritmikus (lásd 15.50. ábra), szemilogaritmikus (vagyis a vízszintes tengelyen logaritmikus skálázású) és duplán logaritmikus (lásd 15.51. ábra) grafikonokat készíthetünk.

```
> # use a random number generator for uniform distribution
> noise := stats[ random, uniform[0,0.1] ]:
> plotpoints := [ seq( [ 0.2*i,
> exp(0.2*i) + noise() ], i=0..20 ) ]:
>   logplot( plotpoints, style=point, symbol=circle );
```



15.50. ábra: Logaritmikus grafikon

```
> loglogplot( x^3 + exp(-x), x=1/10..50, numpoints=200,
>   tickmarks=[3,4], labels=['x','x^3 + exp(-x)'],
>   scaling=constrained, axes=frame );
```



15.51. ábra: Duplán logaritmikus grafikon

Meggyőződhetünk róla, hogy „jó" ábra eléréséhez az utóbbi esetben szükséges a `numpoints` opció.

A villamosmérnökök általában a „Bode-féle ábrázolásnak" nevezett szemilogaritmikus megjelenítéssel dolgoznak. Tegyük föl, hogy valamely elektromos áramkörben adott frekvenciatartomány esetén a következő válaszfüggvény szerinti `Vout` feszültség keletkezik:

```
> alias( I=I, J=sqrt(-1) ):
> Vout := 1/(-12*J*omega^3-2*omega^2+7*J*omega+1);
```

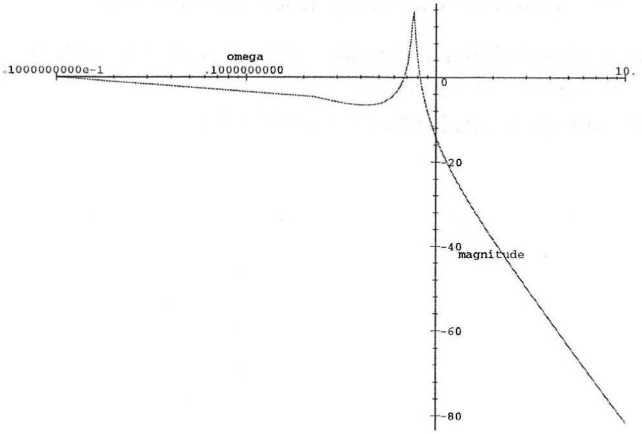
$$Vout := \frac{1}{-12\,J\,\omega^3 - 2\,\omega^2 + 7\,J\,\omega + 1}$$

A Bode-féle ábrázolásnál a kimenő feszültség nagyságát decibelben (vagyis a $20\,{}^{10}\log x$-nek megfelelő egységekben) mérjük, és szemilogaritmikus skálázással rajzoljuk föl. (Lásd 15.52. ábra.)

```
> magnitude := 20*log[10]( evalc(abs(Vout)) );
```

$$magnitude := 20\,\frac{\ln\left(\frac{1}{\sqrt{-164\,\omega^4+45\,\omega^2+1+144\,\omega^6}}\right)}{\ln(10)}$$

```
> M := semilogplot( magnitude, omega=0.01..10,
>   labels=['omega ','magnitude '] ):
> M; # display magnitude
```



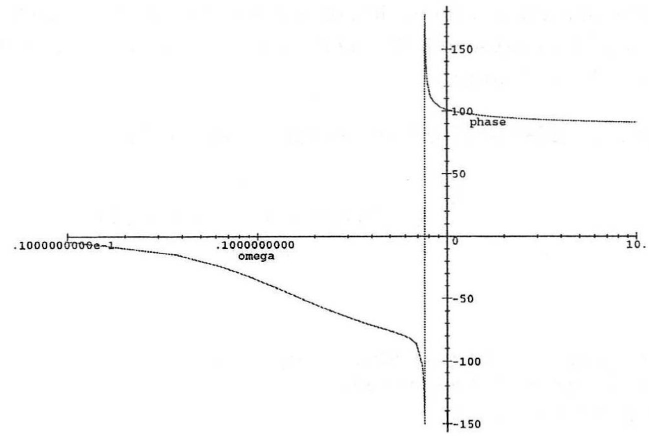
15.52. ábra: A feszültség nagyságának Bode-féle ábrázolása

A Bode-féle ábrázolásnál a kimenő feszültség fázisszögét, más néven argumentumát is föltüntetik. (Lásd 15.53. ábra.)

```
> phase := evalc( argument(Vout) );
```

$$phase := \arctan\left(-\frac{-12\,\omega^3+7\,\omega}{(-2\,\omega^2+1)^2+(-12\,\omega^3+7\,\omega)^2}, \frac{-2\,\omega^2+1}{(-2\,\omega^2+1)^2+(-12\,\omega^3+7\,\omega)^2}\right)$$

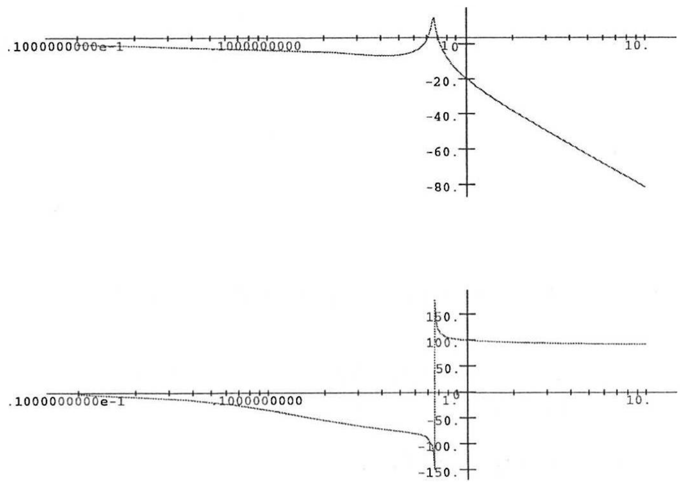
```
> P := semilogplot( 180/Pi*phase, omega = 0.01..10,
>   labels=['omega ','phase '] ):
> P; # display phase
```



15.53. ábra: A fázisszög Bode-féle ábrázolása

A Bode-féle ábrázolás valójában a két előző grafikon egymás alá helyezett, együttes megjelenítését jelenti (15.54. ábra).

```
> display( array(1..2,1..1,[[M],[P]]) );
```

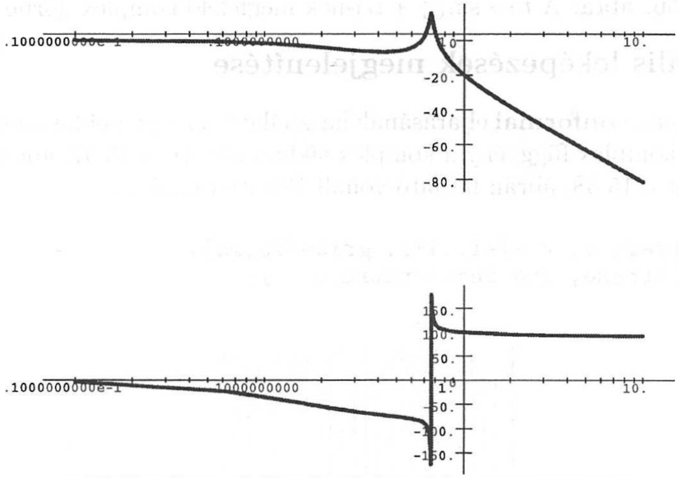


15.54. ábra: Bode-féle ábrázolás

Az előző parancsokat összefoglalhatjuk egy eljárásba is. Az alábbiakban olyan **Bodeplot** nevű eljárást definiálunk, amely a plotopciók megfelelő beállítását is elvégzi (v. ö. 15.55. ábra).

```
> Bodeplot := proc( voltage, domain )
>   local M, P, l, magnitude, omega, opts,
>       phase, r, toption;
>   if not typematch( domain,
>     omega::name = l::algebraic..r::algebraic) then
```

```
>       ERROR(‘invalid input‘)
>     elif not type( voltage, ratpoly(anything,omega)) then
>       ERROR(‘input must be a rational function in‘,omega)
>     fi;
>     opts := [ args[3..nargs] ];
>     if not hasoption( opts, ’thickness’, toption, ’opts’ )
>     then toption:=2 # default thickness
>     fi;
>     l := evalf(l); r := evalf(r);
>     M := plots[semilogplot](
>       20*log[10]( evalc(abs(voltage)) ),
>       omega=l..r, ’thickness’=toption,
>       labels=[omega,magnitude], op(opts) );
>     P := plots[semilogplot](
>       180/Pi*evalc(argument(voltage)),
>       omega=l..r, ’thickness’=toption,
>       labels=[omega,phase], op(opts) );
>     plots[display]( array(1..2,1..1,[[M],[P]]));
>   end:
>   Bodeplot( Vout, omega=0.01..10, thickness=5,
>     color=blue, numpoints=200); # greater thickness
```



15.55. ábra: A **Bodeplot**–tal készített grafikon

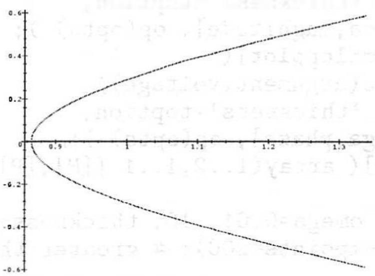
Néhány megjegyzést is fűzünk ehhez a kis grafikai programhoz. Az első föltételes utasítással az input helyességét ellenőrizzük. Ezután az opcionális argumentumokat tároljuk az `opts` nevű változóban, és azt is megnézzük, hogy szerepelt-e a `thickness` opció. Ezt a **hasoption** eljárással végezzük el. Ha megadtuk a `thickness` opciót, a **hasoption** gondoskodik megfelelő kezeléséről. Amennyiben nem fordult elő ez az opció, az alapföltételezés szerinti 2 értéket vesszük. Maple eljárásoknak így szokás opciókat és alapföltételezés szerinti értékeket átadni.

```
>   alias( J=J, I=sqrt(-1) ): # I = sqrt(-1) again
```

Komplex görbék

Komplex görbék, vagyis az ℝ-ből ℂ-be leképező függvények a **complexplot** eljárással ábrázolhatók. Elegendő lesz egyetlen példa (lásd 15.56. ábra). Emlékeztetjük az Olvasót, hogy I= $\sqrt{-1}$.

```
> complexplot( sin( Pi/3 + t*I ), t = -1..1 );
```

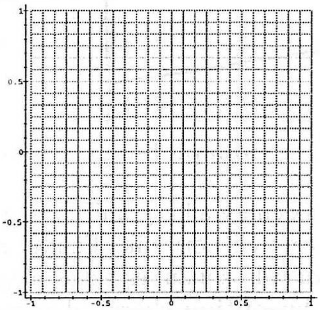


15.56. ábra: A $t \mapsto \sin(\frac{\pi}{3} + ti)$-nek megfelelő komplex görbe

Konformális leképezések megjelenítése

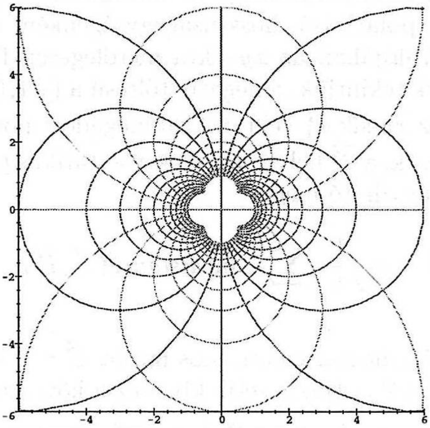
A `plots` csomag **conformal** eljárásának használatára is egy példát ismertetünk. A $z \longmapsto 1/z$ komplex függvény a komplex síkban fölvett, a 15.57. ábrán látható négyzetrácsot a 15.58. ábrán látható vonalhálózatba viszi át.

```
> conformal( z, z=-1-I..1+I, grid=[25,25],
>   axes=frame, scaling=constrained );
```



15.57. ábra: Négyzetrács a komplex síkban

```
> conformal( 1/z, z=-1-I..1+I, grid=[25,25],
>   numxy=[100,100], axes=frame,
>   scaling=constrained, view=[-6..6,-6..6]);
```



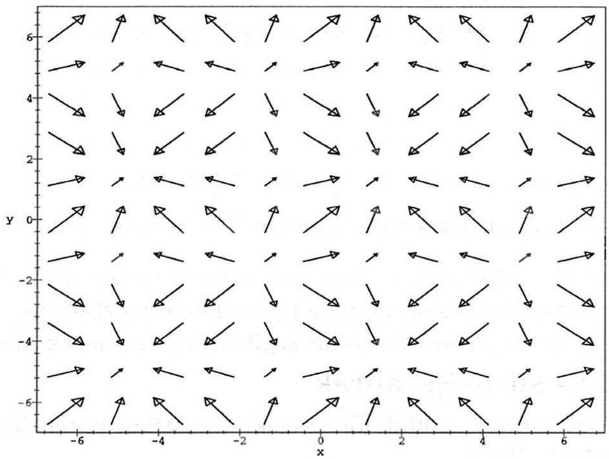
15.58. ábra: A $z \mapsto \frac{1}{z}$ komplex függvény konformális ábrázolása

A 15.58. ábra az eletromos dipólus kétdimenziós megfelelőjével, vagyis nagyon közeli párhuzamos egyeneseken elhelyezett, két ellentétes töltés által létrehozott elektromos mező képével kapcsolatos.

Vektormezők ábrázolása

A **fieldplot** eljárást kétdimenziós vektormezők fölrajzolására használjuk. Első példánk a következő (lásd 15.59. ábra):

```
> fieldplot( [cos(x),cos(y)], x=-2*Pi..2*Pi, y=-2*Pi..2*Pi,
>   arrows=SLIM, grid=[11,11], axes=box );
```



15.59. ábra: A $[\cos x, \cos y]$ mező képe a $[-2\pi, 2\pi] \times [-2\pi, 2\pi]$ négyzeten

A gradiensmező megjelenítésének példájaként azt az elektromos mezőt fogjuk ábrázolni, amely két egységnyi távolságú párhuzamos egyeneseken egyenletesen elhelyezkedő ellentétes polaritású, hosszúságegységenként 1 egységnyi töltések hatására keletkezik. Valójában az xy-síkra merőlegesen fölvett töltéshordozó egyeneseket végtelennek tekintjük, a negatív töltésű a $(-1,0)$, a pozitív az $(1,0)$ ponton halad át. Ha az xy-sík $\vec{P_i}$ vektorokkal megadott pontjain haladnak át a töltéshordozó l_i egyenesek, a $\vec{P}$ helyen a ϕ elektrosztatikus potenciál a következő általános képlettel határozható meg:

$$\phi = -\frac{1}{2\pi\epsilon_0}\sum_i l_i \ln(distance(\vec{P},\vec{P_i})).$$

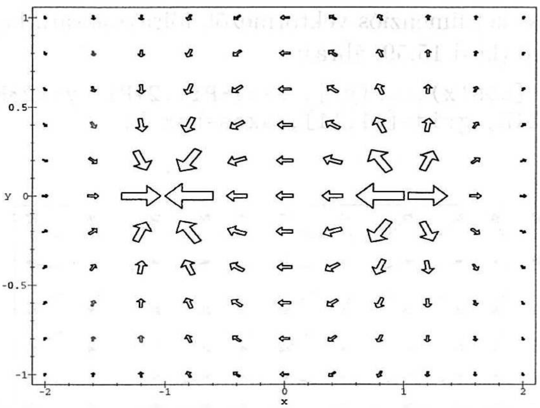
A töltéseloszlás által létrehozott elektromos mezőt $\vec{E} = -\nabla\phi$ definiálja. Ha az $\frac{1}{2\pi\epsilon_0}$ skálafaktort 1-nek választjuk, esetünkben ez a következőt jelenti:

```
> phi := ln( sqrt((x+1)^2+y^2) ) - ln( sqrt((x-1)^2+y^2) );
```

$$\phi := \ln(\sqrt{x^2+2\,x+1+y^2}) - \ln(\sqrt{x^2-2\,x+1+y^2})$$

A kétdimenziós gradiens vektor a következő módon számítható és ábrázolható (lásd 15.60. ábra):

```
> gradplot( -phi, x=-2..2, y=-1..1,
>   arrows=THICK, grid=[11,11], axes=box );
```



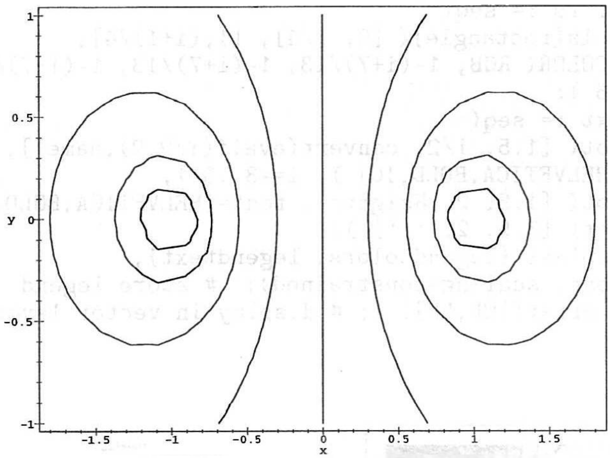
15.60. ábra: Egyenes töltéseloszlás eletromos tere

A 17. fejezetben a `DEtools` csomag tárgyalásánál még visszatérünk erre a példára. A megfelelő ekvipotenciálisok, vagyis a konstans eletrosztatikus potenciálnak megfelelő görbék a szintvonalas ábrázolás segítségével kaphatók meg.

Szintvonalas és sűrűségi ábrák

Előző példánkat folytatva rajzoljuk föl az elektrosztatikus potenciálhoz tartozó szintvonalakat (15.61. ábra):

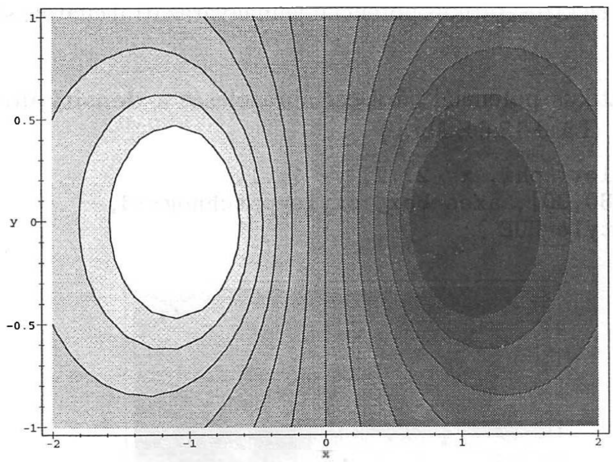
```
> contourplot( phi, x=-2..2, y=-1..1, color=black,
>   numpoints=500, axes=box, contours=10 );
```



15.61. ábra: Az elekrosztatikus potenciál ekvipotenciálisai

Azt is megadhatjuk, hogy milyen szinteket kívánunk az ábrán szerepeltetni, sőt az egyes szintek színezését is előírhatjuk (az alábbi rajzhoz szürkeárnyalatokat választottunk, v. ö. 15.62. ábra).

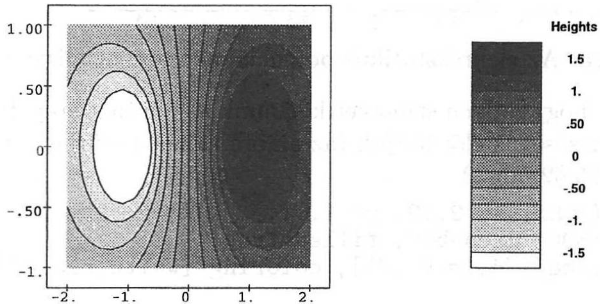
```
>  contourplot( phi, x=-2..2, y=-1..1,
>    numpoints=500, axes=box, filled=true,
>    contours=[seq(i/4,i=-6..6)], coloring=[white,black] );
```



15.62. ábra: Az elektrosztatikus potenciál megadott szinteknek megfelelő ekvipotenciálisai

Kis extra munkaráfordítással az egyes szürkeségi szintek jelentését megadó magyarázó föliratot is készíthetünk. Megmutatjuk, hogy viszonylag egyszerű eszközökkel mit érhetünk el:

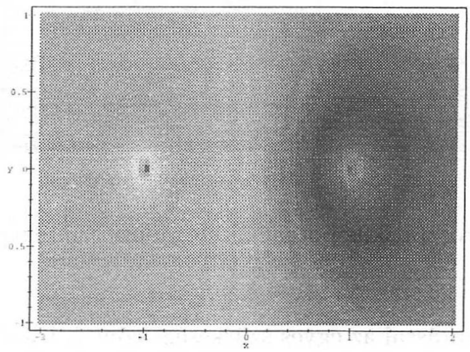
```
> CP := ": # store contour plot
> legendcolors := seq(
>   plottools[rectangle]( [0, i/4], [1,(i+1)/4],
>   color=COLOR( RGB, 1-(i+7)/13, 1-(i+7)/13, 1-(i+7)/13) ),
>   i=-7..6 ):
> legendtext := seq(
>   textplot( [1.5, i/2, convert(evalf(i/2,2),name)],
>   font=[HELVETICA,BOLD,10] ), i=-3..3 ),
>   textplot( [1.5, 2, Heights], font=[HELVETICA,BOLD,10] ),
>   textplot( [3.5, 2, ‘ ‘] ):
> LP := display( {legendcolors, legendtext},
>   axes=none, scaling=constrained): # store legend
> display( array([CP,LP]) ); # display in vector layout
```



15.63. ábra: Az elektrosztatikus potenciál jelmagyarázattal ellátott szintvonalas ábrája

A ϕ elektrosztatikus potenciál sűrűségi ábrázolását a **densityplot** eljárással hozhatjuk létre. (Lásd 15.64. ábra.)

```
> densityplot( phi, x=-2..2, y=-1..1,
>   grid=[80,80], axes=box, style=patchnogrid,
>   colorstyle=HUE );
```



15.64. ábra: Az elektrosztatikus potenciál finom rácsbeosztással készült sűrűségi ábrája

15.6. Síkgeometria

A `geometry` csomagban az euklideszi síkgeometria tanulmányozását elősegítő eszközöket találunk. A csomag geometriai és grafikai lehetőségeiről elegendő képet ad a következő két példa.

Apollonius tétele. *Tetszőleges derékszögű háromszögben a három oldalfelező ponton áthaladó kört megrajzolva a magasságok talppontjai is ugyanezen a körön lesznek.*

Lássuk, hogyan illusztrálható és „bizonyítható" a tétel a `geometry` csomag segítségével. Először töltsük be a csomagot, és definiáljuk az OAB háromszöget:

```
> with(geometry): # load geometry package
> assume( x>0, y>0 ): # assume positive lengths
> triangle( T,
>   [ point(O,0,0), point(A,x,0), point(B,0,y) ] ):
```

Ezután adjuk meg az oldalfelező pontokat és az általuk meghatározott kört:

```
> midpoint( D, O, A ):
> midpoint( E, A, B ):
> midpoint( F, O, B ):
> circle( C, [ D, E, F ] ):
```

Legyen G az O csúcshoz és az AB oldalhoz tartozó magasság talppontja:

```
> projection( G, O, line('dummy',[A,B]) ):
> line( L, [O,G] ): # altitude
```

A `geometry` csomag legfontosabb képessége, hogy olyan geometriai kérdéseket is föltehetünk, mint például „Rajta van-e ez a pont az egyenesen?" vagy „Merőleges-e a két egyenes?". A tétel helyességét tehát úgy ellenőrizhetjük, hogy megkérdezzük a Maple-től, vajon rajta van-e a G pont a C körön.

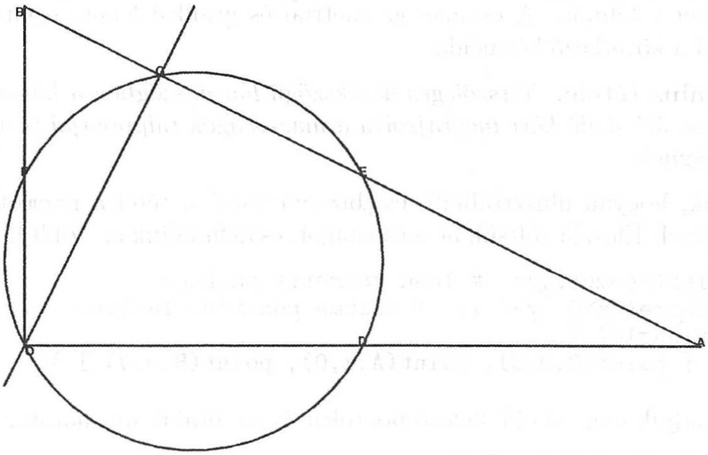
```
> IsOnCircle( G, C );
```

$$true$$

Válasszunk ki egy meghatározott T háromszöget, és rajzoljuk föl a megfelelő ábrát:

```
> assign(x,2): assign(y,1):
> draw( [ C(printtext=false), T, D, E, F, G, L ],
>   axes=none, thickness=3, font=[HELVETICA,BOLD,10],
>   color=blue, scaling=constrained );
```

Ahogy a fönti parancsból is látható, a `geometry` csomagban elérhető objektumok grafikus megjelenítését a **draw** eljárás végzi. Mind az egyes objektumokra vonatkozó lokális, mind globális plotopciókat alkalmazhatunk. A következő példában az ábra csinosítására erősen ki is használjuk ezt a sajátosságot. Most csak annyit mondtunk a Maple-nek, hogy az Apollonius kör középpontját ne rajzolja meg.



15.65. ábra: Apollonius tételének illusztrációja

A fönti tétel algebrai jellegű levezetését az [51] 6.4. alfejezetének 3. példája tartalmazza.

Simpson tétel. *A háromszög körülírt körének tetszőleges pontjából a három oldalegyenesre bocsátott merőlegesek talppontjai egy egyenesre esnek.*

Lássuk, hogyan tudjuk grafikusan illusztrálni és „bizonyítani" ezt a tételt. Először az OAB háromszöget és C körülírt körét definiáljuk:

```
> triangle( T,
>   [ point(O,0,0), point(A,3,0), point(B,2,2) ] ):
> circumcircle( C, T ):
```

Ezután a kör egy D pontját adjuk meg. Olyan föltételt kell találnunk, amely azt írja le, hogy egy tetszőleges (u, v) pont mikor van rajta a C körön:

```
> point( D, u, v ):
> IsOnCircle( D, C, 'condition' );

IsOnCircle:
hint: unable to determine if u^2+v^2-3*u-v is zero
```

$$FAIL$$

```
> condition;
```

$$u^2 + v^2 - 3\,u - v = 0$$

Tegyük föl, hogy teljesül ez a föltétel:

```
> assume(condition):
```

A D pontból az OA, OB és az AB oldalegyenesekre bocsátott merőlegesek E, F, illetve G talppontját a következő parancsokkal kaphatjuk meg:

```
> projection( E, D, line( 'dummy', [O,A] ) ):
> projection( F, D, line( 'dummy', [O,B] ) ):
> projection( G, D, line( 'dummy', [A,B] ) ):
```

Igazoljuk Simpson tételét:

```
> AreCollinear( E, F, G );
```

$$true$$

Egy adott T háromszögre tehát a Maple be tudja bizonyítani a Simpson-tételt! Válasszunk egy rögzített D pontot a körön:

```
> solve( condition );
```

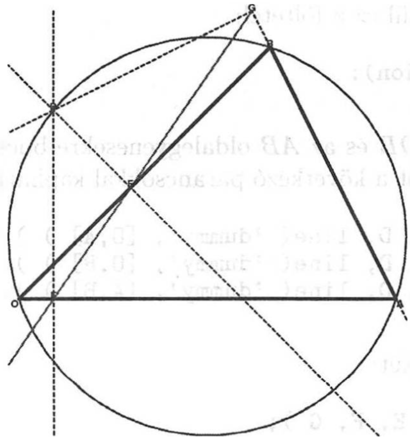
$$\{v\tilde{} = v\tilde{},\, u\tilde{} = \frac{3}{2} + \frac{1}{2}\sqrt{9 - 4\,v\tilde{}^{2} + 4\,v\tilde{}}\},$$
$$\{v\tilde{} = v\tilde{},\, u\tilde{} = \frac{3}{2} - \frac{1}{2}\sqrt{9 - 4\,v\tilde{}^{2} + 4\,v\tilde{}}\}$$

A megoldás alapján könnyen ellenőrizhető, hogy az $(u, v) = \left(\frac{3}{2} - \frac{1}{2}\sqrt{6}, 1\right)$ pont a körön van. Vegyük ezt a pontot, és határozzuk meg a D-t az oldalegyeneseken lévő talppontokkal összekötő egyeneseket, a talppontokon átmenő L egyenest, valamint az ábrázolás javításához szükséges további segédegyenest:

```
> assign( u, 3/2-1/2*sqrt(6) ): assign(v,3/2):
> line( DE, [D,E] ): line( DF, [D,F] ):
> line( DG, [D,G] ): line(L,[E,G]):
> line(H,[B,G]):
```

Minden készen áll a Simpson-tételt illusztráló ábra elkészítéséhez. (V. ö. 15.66. ábra.)

```
> draw(
>   [ C(filled=false, linestyle=1, printtext=false),
>     T(color=black, filled=false, thickness=5, linestyle=1),
>     DE, DF, DG, H, D, L(color=red,linestyle=1),
>     E, F, G ],
>   axes=none, linestyle=2, thickness=3, color=blue,
>   scaling=constrained, font=[HELVETICA,BOLD,10]
> );
```



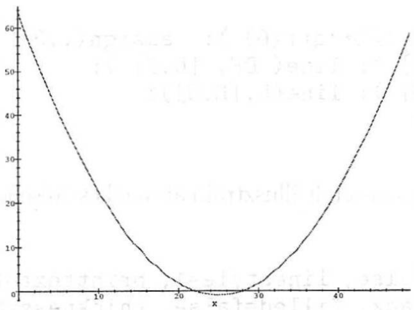
15.66. ábra: A Simpson-tétel illusztrációja

Gyakorlásként bizonyítsuk be a `geometry` csomag segítségével Simpson tételét tetszőleges háromszögre! A tétel teljesen algebrai jellegű tárgyalása megtalálható [119]-ben.

15.7. Grafikonok álruhában

Az adaptív kétdimenziós ábrázolás többnyire elfogadható grafikonokat eredményez már minimális számú mintapont fölhasználásával is. De ügyelnünk kell arra, hogy valóban megfelelő-e a mintavétel. A megtévesztő ábrák extrém eseteként ábrázoljuk az $x \longmapsto \frac{1}{10}(x-25)^2 + \cos(2\pi x)$ függvényt a $(0, 49)$ intervallumon. (Lásd 15.67. ábra.)

```
> plot( (x-25)^2/10 + cos(2*Pi*x), x=0..49 );
```

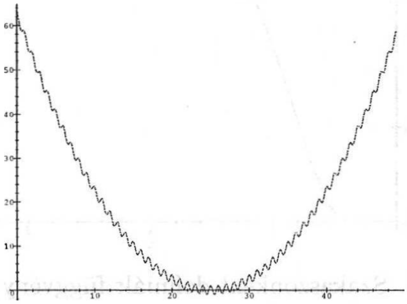


15.67. ábra: Az $x \longmapsto \frac{1}{10}(x-25)^2 + \cos(2\pi x)$ függvény megtévesztő ábrázolása

A fölrajzolt (hibás!) grafikon egészen közel áll az $y = \frac{1}{10}(x-25)^2$ parabolájához. Ezt a hamis látszatot az okozta, hogy az alapértelmezésnek megfelelően válasz-

tott 49 közel ekvidisztáns mintapontban a kiszámított függvényértékek valóban közel esnek a parabolához. Mivel a Maple-nek semmi oka azt föltételezni, hogy itt valami nincs rendben, nem is próbálkozik nagyobb vagy más eloszlású mintával. Nekünk kell eldönteni, hogy akarunk-e több mintapontot (a `numpoints` opció beállításával) vagy más ábrázolási tartományt.

```
> plot( (x-25)^2/10+cos(2*Pi*x), x=0..49, numpoints=2000 );
```



15.68. ábra: Az $x \longmapsto \frac{1}{10}(x-25)^2+\cos(2\pi x)$ kijavított grafikonja

15.8. Egy gyakori féreértés

Ha a **plot**($f(x)$, $x = x_{\min} .. x_{\max}$) paranccsal arra kérjük a Maple-t, hogy ábrázolja az x változó f függvényét, akkor a rendszer először kiértékeli $f(x)$-et, s valószínűleg valamely x-et is tartalmazó szimbolikus kifejezést kap eredményül. Ezután a kapott kifejezést már numerikusan értékeli ki a választott mintapontokban. Ebből gondjaink származhatnak, ha például szakaszonként definiált vagy numerikus függvényt próbálunk fölrajzoltatni. Az alábbiakban erre mutatunk példát, s rögtön négy lehetőséget is említünk a problémák orvoslására. Tegyük föl, hogy az

```
> f := t -> if t>0 then exp(-1/t^2) else 0 fi:
```

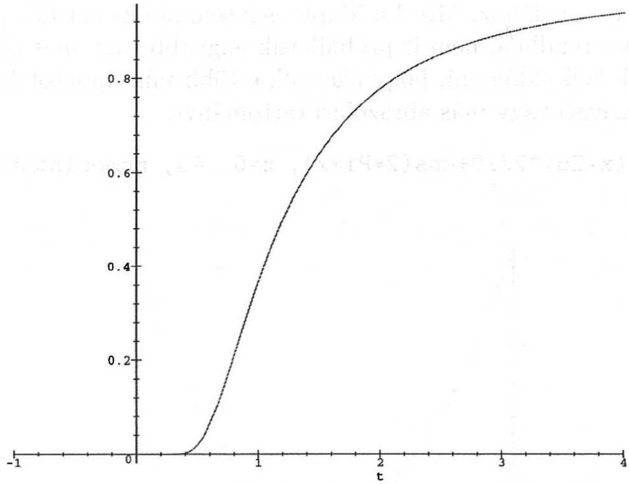
paranccsal definiált f függvényt szeretnénk ábrázolni:

```
> plot( f(t), t=-1..4 );

Error, (in f) cannot evaluate boolean
```

Látható, hogy grafikon helyett csak hibaüzenetet kaptunk. Ezt az okozta, hogy $f(t)$ kiértékelésekor a Maple nem tudta eldönteni, hogy a $t > 0$ föltétel teljesül-e. A szokásos trükk ilyenkor aposztrófok használata a túl korai kiértékelés elkerülésére:

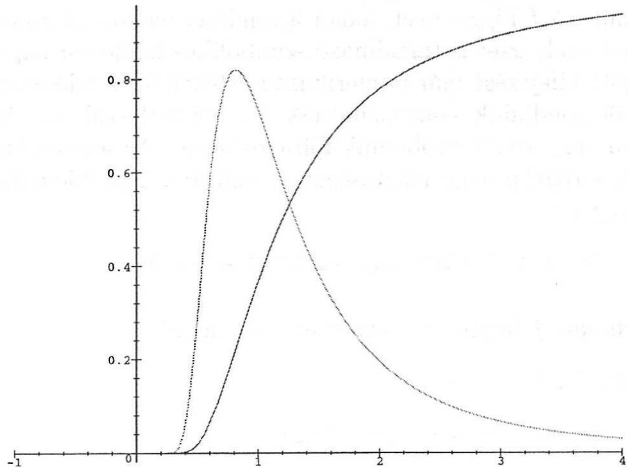
```
> plot( 'f(t)', t=-1..4 );
```



15.69. ábra: Szakaszonként definiált függvény grafikonja

Következő lehetőségünk a függvényeknek megfelelő jelölésmód alkalmazása. Például az f függvényt és f' deriváltját közös ábrán jeleníti meg a következő parancs. (Lásd 15.70. ábra.)

```
> plot( {f,D(f)}, -1..4 );
```



15.70. ábra: Szakaszonként definiált függvénynek és deriváltjának grafikonja

Másik két megoldásunk a függvény definíciójának megváltoztatásán alapul. Ha már a Maple eleve rendelkezik a szakaszonként definiált függvényeket helyesen kezelő **piecewise** eljárással, miért ne használnánk ezt?

```
> f := t -> piecewise( t>0, exp(-1/t^2), 0 ):
> f(t); # no problem
```

$$\begin{cases} e^{(-\frac{1}{t^2})} & 0 < t \\ 0 & otherwise \end{cases}$$

Továbbá, ahogy a 8.3. alfejezetben láttuk, megadhatjuk úgy is a függvény definícióját, hogy gond nélkül elfogadjon nemnumerikus argumentumokat is. Ennek egyik módja lehet

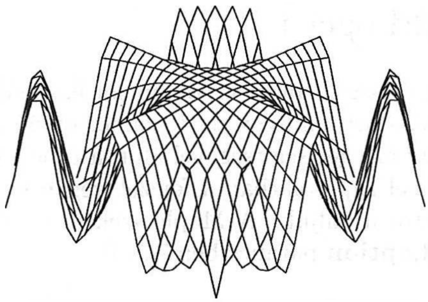
```
> f := t -> if not type(t,numeric) then
>          'procname'(t)
>        elif t>0 then
>          exp(-1/t^2)
>        else 0
>        fi:
> f(t); # no problem
```

$$\mathrm{f}(t)$$

15.9. Néhány egyszerű háromdimenziós ábra

A kétváltozós függvények ábrázolása ugyanolyan könnyen megy, mint az egyváltozósaké. Csupán a **plot3d** Maple eljárást kell alkalmaznunk mindkét változóra megadott értéktartományokkal. Ábrázoljuk például a $z = \cos(xy)$ függvénnyel definiált felületet -3 és 3 közti x és y értékekre. (Lásd 15.71. ábra.)

```
> plot3d( cos(x*y), x=-3..3, y=-3..3, color=black );
```



15.71. ábra: A $z = \cos(xy)$ felület képe a takart részek eltávolításával

A fönti parancs speciális esete az x és y változóktól függő $f(x, y)$ *formulával* definiált felületet fölrajzoló

$$\textbf{plot3d}(\,f(x,y), x = a\,..\,b,\ y = c\,..\,d,\ \textit{options}\,)$$

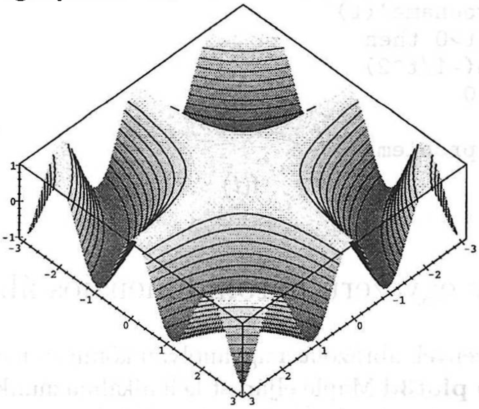
utasításnak, amelyben $a\,..\,b$ és $c\,..\,d$ az x, illetve az y értéktartományát definiálja, *options* pedig a következő alfejezetben vizsgált opciók (esetleg üres) listája.
A kétváltozós f *függvény* grafikonját a

$$\textbf{plot3d}(\,f, a\,..\,b,\ c\,..\,d,\ \textit{options}\,)$$

alternatív szintakszissal kapjuk, ahol $a..b$ és $c..d$ adja meg a két horizontális értéktartományt, *options* pedig nulla vagy több opciót ír le.

Az alábbiakban ismét az $(x, y) \longmapsto \cos(xy)$ függvény grafikonját rajzoljuk föl, de most a függvényes jelölésmódot alkalmazzuk, és néhány opciót is megváltoztatunk:

```
> f := (x,y) -> cos(x*y):
> plot3d( f, -3..3, -3..3, grid=[50,50], axes=box,
>   scaling=constrained, style=patchcontour,
>   shading=zgrayscale );
```



15.72. ábra: Az $(x, y) \mapsto \cos(xy)$ függvény szintvonalakkal kiegészített szürkeárnyalatos grafikonja

15.10. A **plot3d** opciói

A kétdimenziós esethez hasonlóan itt is sok opciót használhatunk ábráink „testreszabására". Az X Window rendszer alatt számos opció a megfelelő menüpont kiválasztásával és/vagy az egér segítségével állítható be. Némely opciókat most tárgyalunk, a többiekkel kapcsolatban a fejezet végén található, a **plot3d** opcióit fölsoroló táblázatot ajánljuk. A Maple szekció során bármikor segítséget kaphatunk a **?plot3d,option** parancs beírásával.

Stílus

A `style` = *displaystyle* opcióval ugyanazt a felületet különböző módokon jeleníthetjük meg. A legegyszerűbb a `point` stílus, amely csak a kiszámított mintapontokat ábrázolja. Ha a `line` stílusopciót választjuk, a kiszámított értékeket egyenes szakaszokkal köti össze a Maple. Felület ábrázolásakor egyébként a mintapontok általában téglalap alakú ekvidisztáns rácshálózat osztáspontjai. Ha a szomszédos adatpontokat egyenes szakaszokkal kötjük össze, az eredmény a felületet közelítő `wireframe` stílusú ábra. Az előző rész első ábrájáról hiányoznak azok a szakaszok, amelyek átlátszatlannak képzelt felület esetén az adott nézőpontból nem láthatók. Ez a Maple alapértelmezés szerinti `hidden` stílusa. Ha a felületet több színben vagy szürkeárnyalatokban kívánjuk ábrázolni, használhatjuk a `patch` stílust. Amennyiben el akarjuk hagyni a rácshálózat képét,

válasszuk a `patchnogrid` opciót. A `contour` stílusopcióval a felület szintvonalait kapjuk. A `patchcontour` opciót, a két megfelelő opció kombinációját illusztrálja az utolsó ábra.

Árnyékolás

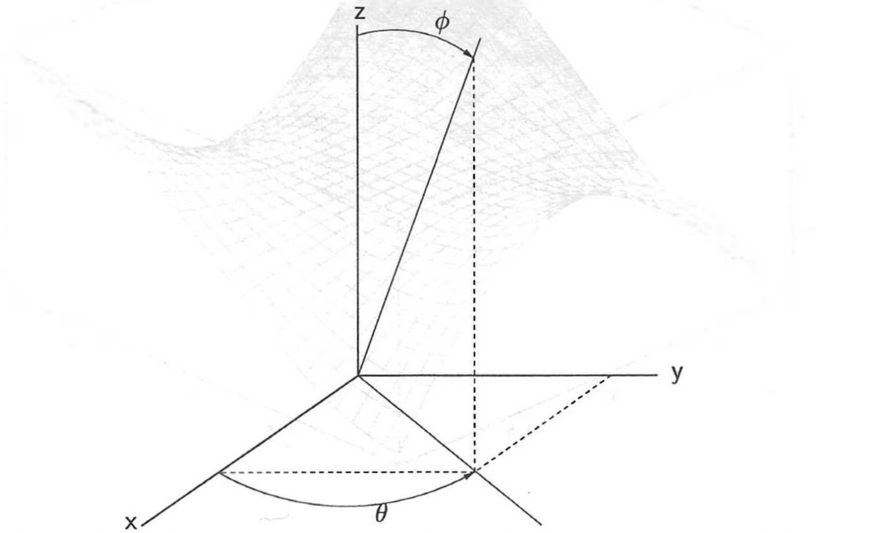
A Maple-ben valamely felület háromféle módon színezhető ki („igazi" színekkel vagy szürkeárnyalatokkal): az `xyz`, az `xy` és a `z` árnyékolás szerint. Amikor a `shading = z` opciót választjuk, a színezés vagy a szürkeségi fokozat a pontok z koordinátája szerint alakul. Az `xy` vagy az `xyz` séma esetén minden koordinátatengelyhez külön színskála tartozik, ezek addíciója határozza meg a felület pontjainak színét. A `zhue` és a `zgrayscale` opciók a z értékektől függő HUE szinmodell szerinti, illetve szürkeskálás grafikonokat eredményeznek. A `shading = none` hatására az egész ábra egyszínű lesz.

Tengelyek

Az `axes`opcióval a tengelyek ábrázolását adhatjuk meg. Négy választásunk van: `none`, `normal`, `box` és `frame`. A nevek magukért beszélnek; csupán annyit jegyzünk meg, hogy a `frame`-nek megfelelő keretezett tengelyeket a felület csúcsai körül úgy rajzolja meg a rendszer, hogy ne metsszék a felületet.

Irányítás és projekció

Háromdimenziós grafikus objektumokat általában projekcióval szokás megjeleníteni a kétdimenziós képernyőn vagy papíron. A projekció középpontját (másképpen a „nézőpontot") az ábra „tartódobozának" (vagyis az ábrát magábafoglaló legkisebb téglatestnek) középpontjába elhelyezett lokális Descartes-féle koordinátarendszerre vonatkoztatott szférikus koordinátákkal adjuk meg. (Lásd 15.73. ábra.)



15.73. ábra: Irányítás megadása 3 dimenzióban a ϕ és a θ szögekkel

A θ és a ϕ szögek az `orientation=` $[\theta, \phi]$ opcióval állíthatók be. A θ forgatási szög növelése az ábrát a z-tengely körül az óramutató járásával ellentétes irányban forgatja el. A 0 érték annak felel meg, hogy a felület elülső oldalát és tetejét látjuk.

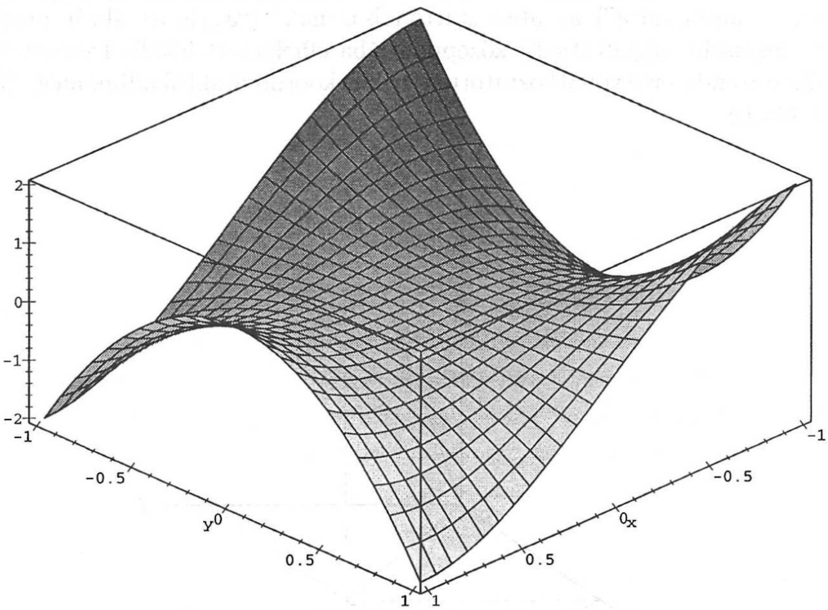
A ϕ függőleges látásszög jelentése: a 0 értéknél függőlegesen lefelé tekintünk a felületre, 90°-nál elölről nézzük (az xy sík egyenesnek látszik), ennél nagyobb értékekre alulról látjuk a felületet.

A `projection = p` opció, ahol p a $[0, 1]$ intervallumba eső valós konstans, a nézőpontnak a felülettől való távolságát adja meg. A 0 (`fisheye`) a két pont egybeesését, 1 (`ortogonal`) végtelen távolságot, a `normal` érték $\frac{1}{2}$-et jelent.

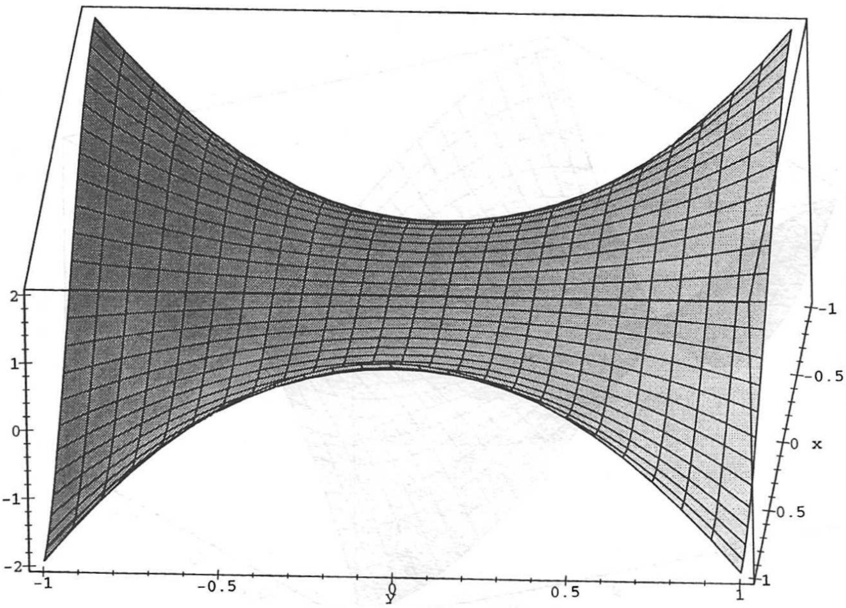
Néhány ábra a szavaknál is többet mond. A továbbiakban olyan rajzokat mutatunk különböző nézőpontokból, amelyeket eredetileg a

```
> plot3d( x^3-3*x*y^2, x=-1..1, y=-1..1,
>    style=patch, axes=box );
```

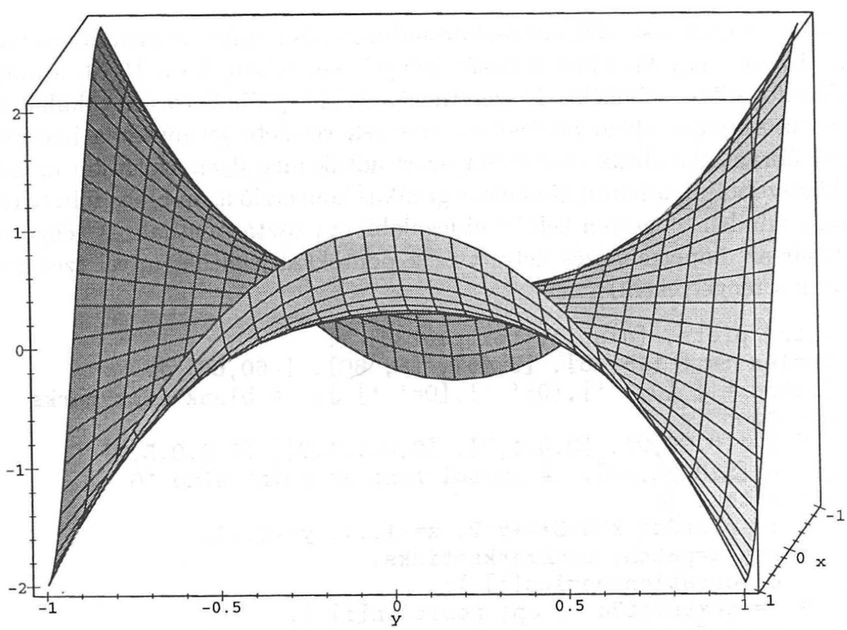
paranccsal hoztunk létre:



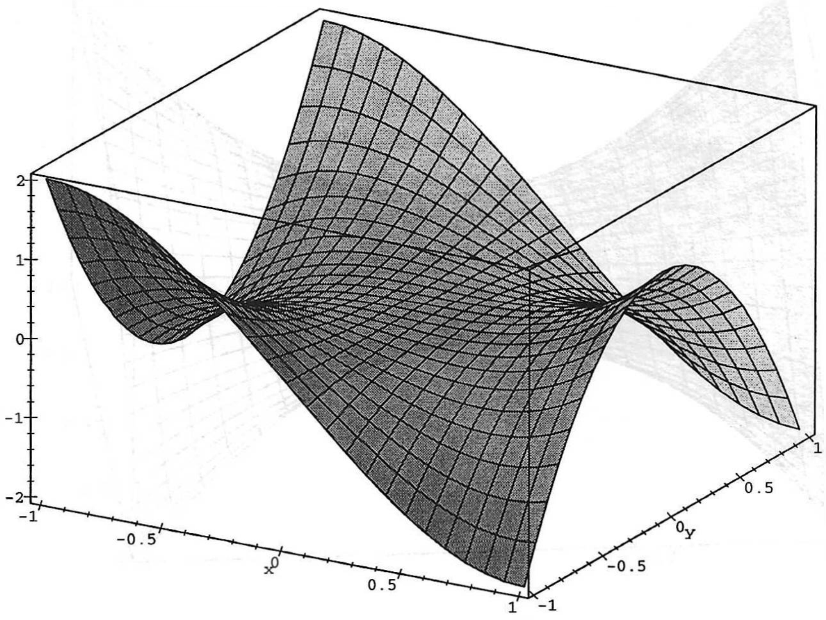
theta = 45, phi = 45



theta = 5, phi = 45



theta = 5, phi = 80

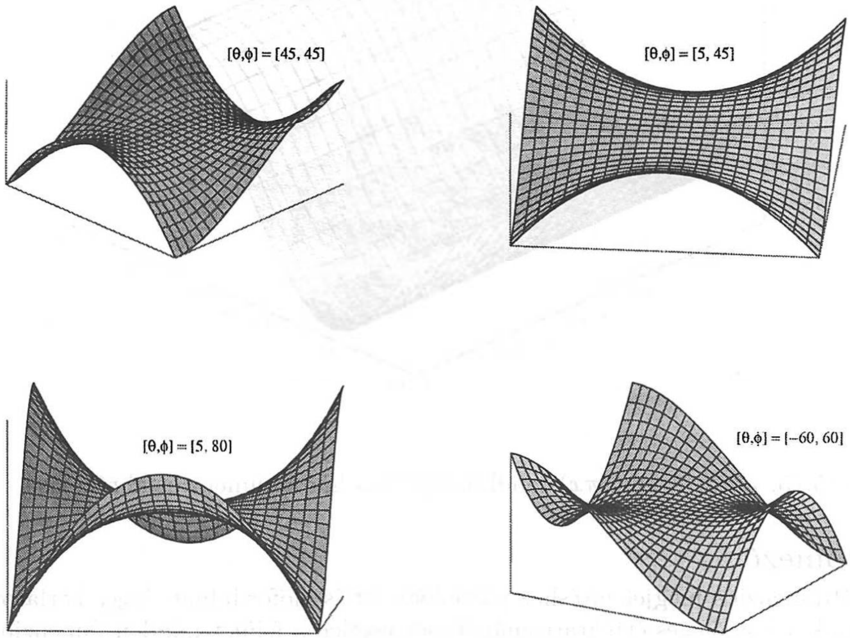


theta = -60, phi = 60

Az előző rajzokból összeállíthatunk háromdimenziós grafikákat tartalmazó tömböt is. Ez még egy kis plusz munkát igényel, különösen, ha a 15.74. ábrához hasonló elfogadható cimkéket is szeretnénk. (Több próbálkozással találhatunk a föliratok számára olyan pozíciókat, amelyek vetülete jól mutat a háromdimenziós ábrán. Az ábrák cimkézését azért adtuk meg ilyen szokatlan módon, hogy kiküszöböljük a háromdimenziós grafikus konverzió azon hibás föltételezését, hogy minden tengelyen kell lenni legalább egy osztáspontnak. A tengelyek beosztását az alapértelmezés helyett csak pontokkal jelöltük, mivel ezek nem látszanak a tengelyeken.)

```
> with( plots, [display,textplot3d] ):
> angles := [ [45,45], [5,45], [5, 80], [-60,60] ]:
> ticks := [ [0=' '],[0=' '],[0=' '] ]:  # blank tick marks
> position :=
>   [ [-1,0.25,2], [0,0.1,3], [0,0.1,1.2], [0.8,0.5,2] ]:
> sf := [SYMBOL,10]:  # symbol font at point size 10
> for i to 4 do
>   P := plot3d( x^3-3*x*y^2, x=-1..1, y=-1..1,
>     style=patch, tickmarks=ticks,
>     orientation=angles[i] ):
>   T := textplot3d( [ op( position[i] ),
>     cat('[q,f] = ',convert(angles[i],name)) ],
>     font=sf, color=blue );
```

```
>    F[i] := display( {P,T}, axes=frame ):
> od:
> plotarray := matrix( 2, 2,
>    [ seq( F[i], i=1..4) ], array ):
> subs( SYMBOL(BOX)=SYMBOL(POINT),
>    display( plotarray ) ); # draw graphics array
```



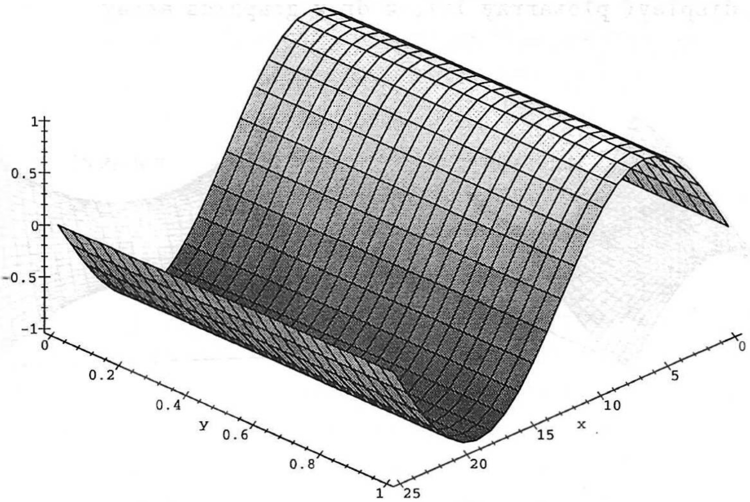
15.74. ábra: Az $x^3 - 3xy^2$ grafikonja különböző nézőpontokból

Megjegyezzük, hogy az árnyékolás vagy a nézőpont megváltoztatásához nincs szükség a grafikus objektum újraszámolására. Ezek csak a képernyőn megjelenő ábra megjelenítési módját befolyásolják. Más a helyzet a következő opció alkalmazásánál.

Rácsméret

A háromdimenziós esetben a Maple-nek nincs a mintapontok által kifeszített rácshálózat finomítására szolgáló algoritmusa. Más szóval a kétdimenziós ábrázolástól eltérően a **plot3d** nem alkalmaz adaptív mintavételi sémát. Az alapföltételezés szerint mindkét dimenzióban 25 ekvidisztáns mintapontot használ föl. Ha az x irányban m, az y irányban pedig n ponttal akarunk dolgozni, a `grid` $=[m, n]$ opcióval változtathatjuk meg a beállításokat. Ügyelnünk kell tehát arra a kellemetlen eshetőségre is, hogy a 15.75. ábrához hasonló megtévesztő háromdimenziós grafikonokat kaphatunk.

```
> plot3d( sin(2*Pi*x), x=0..25, y=0..1,
> axes=frame, style=patch, shading=zgrayscale );
```



15.75. ábra: A $\sin(2\pi x)$ rendkívül gyatra háromdimenziós ábrázolása

Látómező

A kétdimenziós megjelenítéshez hasonlóan itt is előfordulhat, hogy korlátozni akarjuk a függőleges értéktartományt, sőt esetleg a felület minden dimenziójában a megengedett tartományra vonatkozó megszorításokat teszünk. Ehhez a `view` opciót használhatjuk föl. Két lehetséges formája:

$$view = z_{\min} .. z_{\max}$$

és

$$view = \left[x_{\min} .. x_{\max}, y_{\min} .. y_{\max}, z_{\min} .. z_{\max}\right].$$

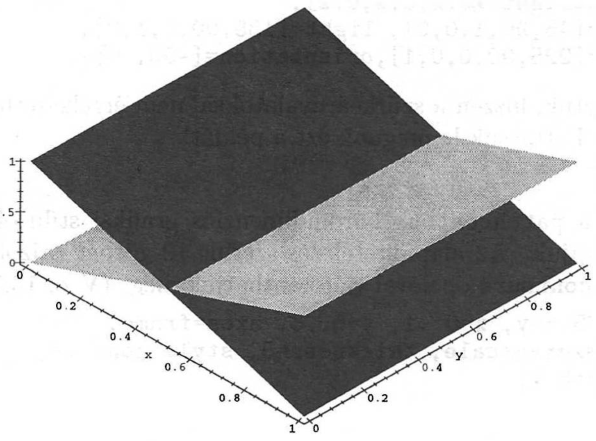
Megvilágítás

A Maple két opció segítségével tud különböző megvilágításokat szimulálni:

- `light` $= [\phi, \theta, r, g, b]$ olyan fényforrást szimulál, amelynek RGB színmodell szerinti összetevői r, g és b erősségűek, továbbá fénye a $[\phi, \theta]$ szférikus koordinátákkal megadott irányból világítja meg a grafikus objektumot.
- `ambientlight` $= [r, g, b]$ az RGB színmodell szerint r, g és b erősségű komponenseket tartalmazó meghatározatlan irányú (szórt) fénynek felel meg.

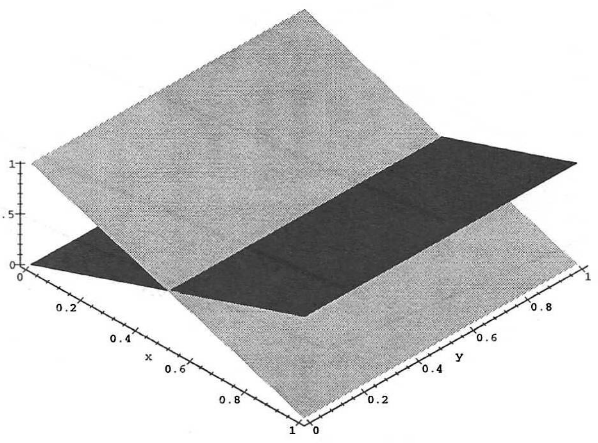
A következő példa, amelyben két metsző síkot az egyikre merőleges irányból szürke fénnyel világítunk meg, jól mutatja a megvilágítás hatását.

```
> plot3d( {x,1-x}, x=0..1, y=0..1, style=patchnogrid,
>   axes=frame, shading=none, orientation=[-45,25],
>   light=[0,0,0.9,0.9,0.9] );
```



15.76. ábra: A felületek megvilágításának hatása (a fény a [0, 0] irányból jön)

```
> plot3d( {x,1-x}, x=0..1,y=0..1, style=patchnogrid,
>   axes=frame, shading=none, orientation=[-45,25],
>   light=[135,0,0.9,0.9,0.9] );
```



15.77. ábra: A felületek megvilágításának hatása (a fény a [135, 0] irányból jön)

A következő példában megadott kód esetén fölülről nézünk egy olyan oktaéderre, amelynek lapjait a három alapszínnek megfelelő három külön fényforrás világítja

meg.

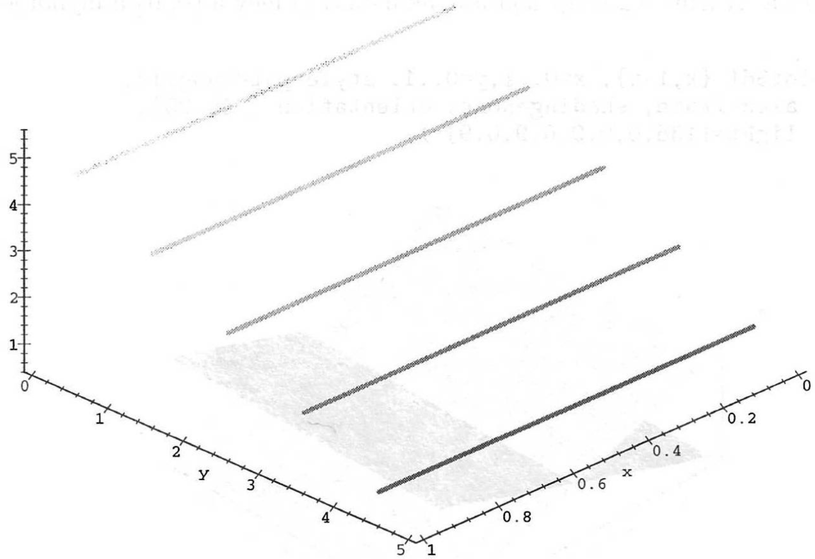
```
> display(
>   plottools[octahedron]([0,0,0],1),
>   style=patch, axes=box,
>   ambientlight=[0.2,0.2,0.2],
>   light=[45,90,1,0,0], light=[135,90,0,1,0],
>   light=[225,90,0,0,1],orientation=[-90, 0]);
```

Az ábrát elhagytuk, hiszen a szürkeárnyalatokkal nem érzékeltethetők a világítási effektusok. Futtassuk le magunk ezt a példát!

Kontúrok

A `contour` és a `patchcontour` háromdimenziós grafikai stílusokkal a felület szintvonalait kapjuk. Az alapföltételezés szerint 10 görbét rajzol föl a Maple, ezt a számot a `contours` opcióval változtathatjuk meg. (V. ö. 15.78. ábra.)

```
> plot3d( 5.5-y, x=0..1, y=0..5, axes=frame,
> shading=zgrayscale, thickness=5, style=contour,
> contours=5 );
```

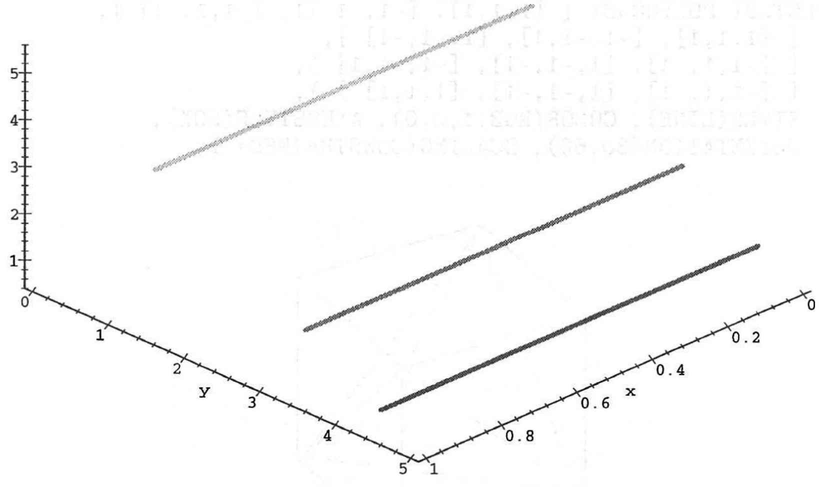


15.78. ábra: A felület szintvonalai

Azt is megadhatjuk explicit módon, hogy mely szintekhez tartozó görbéket szeretnénk fölrajzoltatni. (Lásd 15.79. ábra.)

```
> plot3d( 5.5-y, x=0..1, y=0..5, axes=frame,
>   shading=zgrayscale, thickness=5, style=contour,
```

```
> contours=[1,2,4] );
```



15.79. ábra: A felület adott értékeknek megfelelő szintvonalai

A `plots` csomag **setoptions3d** eljárásával megtekinthetjük az opciók értékét, de be is állíthatjuk őket az alapértelmezésnek vagy a további **plot3d** hívásoknak megfelelő értékekre.

A fejezet végén megtalálhatja az Olvasó a rendelkezésére álló összes háromdimenziós plotopció fölsorolását. Közülük néhányról még nem volt szó, de ezeket is használni fogjuk a fejezet későbbi példáiban. A munkalapos fölhasználói felület lehetővé teszi a `style`, `linestyle`, `thickness`, `symbol`, `gridstyle`, `shading`, `light`, `axes` és a `projection` opciók interaktív beállítását. A többieket valamelyik plot paranccsal tudjuk változtatni.

15.11. Háromdimenziós grafikai struktúrák

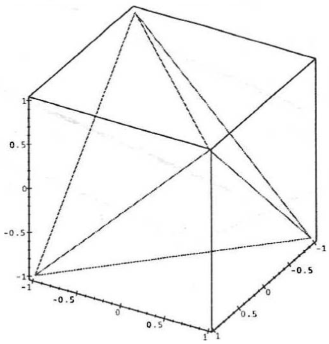
A háromdimenziós ábrák elkészítése a kétdimenziósak generálásához hasonlóan két lépésben történik. Az első fázisban a mintapontokat számítja ki a rendszer, s egy PLOT3D objektumban helyezi el őket. A második fázisban ezt az objektumot jeleníti meg a képernyőn. Ebben a részben két olyan példát hozunk háromdimenziós grafikus objektumokra, amelyek más esetekben prototípusokként szolgálhatnak.

A Maple-ben egy háromdimenziós grafikus objektum a **PLOT3D** eljárás olyan hívásának felel meg, amelynek argumentumai a tengelyeket, az ábrázo-

landó függvényt, a kiszámított mintapontokat, az ábrázolás stílusát, a rácsbeosztást, a színezést stb. adják meg.

A következő objektum az $(1,1,1), (-1,-1,1), (-1,1,-1)$ és az $(1,1,-1)$ csúcsok által meghatározott tetraéder éleit írja le. (Lásd a 15.80. ábrát.)

```
> PLOT3D( POLYGONS( [ [1,1,1], [-1,-1,1], [-1,1,-1] ],
>    [ [1,1,1], [-1,-1,1], [1,-1,-1] ],
>    [ [-1,1,-1], [1,-1,-1], [-1,-1,1] ],
>    [ [-1,1,-1], [1,-1,-1], [1,1,1] ] ),
>    STYLE(LINE), COLOR(RGB,1,0,0), AXESSTYLE(BOX),
>    ORIENTATION(30,60), SCALING(CONSTRAINED) );
```



15.80. ábra: A grafikus primitívekkel fölrajzolt tetraéder

A `POLYGONS(..)`, `STYLE`, `COLOR`, `AXESSTYLE`, `ORIENTATION` és `SCALING` argumentumok a következő hívással létrehozott grafikus objektum részei.

```
> P := polygonplot3d( [
> [ [1,1,1], [-1,-1,1], [-1,1,-1] ],
>   [ [1,1,1], [-1,-1,1], [1,-1,-1] ],
>   [ [-1,1,-1], [1,-1,-1], [-1,-1,1] ],
>   [ [-1,1,-1], [1,-1,-1], [1,1,1] ] ],
>   style=line, color=red, axes=box,
>   orientation=[30,60], scaling=constrained):
> lprint( P ); # print 3D plot data structure

PLOT3D(POLYGONS([[1., 1., 1.], [-1., -1., 1.],
   [-1., 1., -1.]],[[1., 1., 1.], [-1., -1., 1.],
   [1., -1., -1.]],[[-1., 1., -1.], [1., -1., -1.],
   [-1., -1., 1.]],[[-1., 1., -1.], [1., -1., -1.],
   [1., 1., 1.]]),ORIENTATION(30.,60.), STYLE(LINE),
   SCALING(CONSTRAINED), AXESSTYLE(BOX),
   COLOUR(RGB,1.00000000,0,0))

> P; # display the graphics object
```

A rajzot elhagytuk, hiszen ugyanaz, mint a 15.80. ábra.

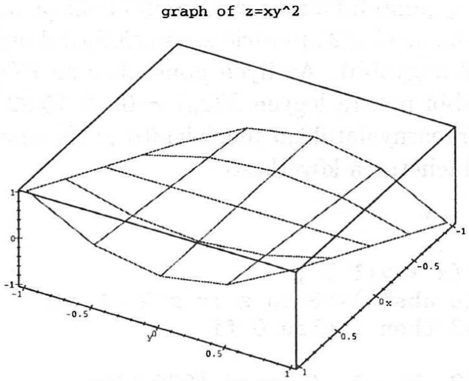
Az $(x,y) \longmapsto xy^2$ függvény grafikonjának ábrázolásához tartozó plotstruktúra általánosabb grafikus objektumra ad példát. (Lásd 15.81. ábra.)

```
> P := plot3d( x*y^2 , x=-1..1, y=-1..1, grid=[5,5],
> axes=box, orientation=[30,30], style=line,
> color=red, title=`graph of z=xy^2` ):
> lprint( P ); # print the 3D plot data structure

PLOT3D(GRID(-1. .. 1.,-1. .. 1.,
    [[-1., -.2500000000000000, 0, -.2500000000000000, -1.],
    [-.5000000000000000, -.1250000000000000, 0,
    -.1250000000000000, -.5000000000000000], [0, 0, 0, 0, 0],
    [.5000000000000000, .1250000000000000, 0,
    .1250000000000000, .5000000000000000],
    [1., .2500000000000000, 0, .2500000000000000, 1.]],
    COLOR(RGB,1.00000000,0,0)),ORIENTATION(30.,30.),
    STYLE(LINE), AXESSTYLE(BOX),AXESLABELS(x,y,``),
    TITLE(`graph of z=xy^2`))
```

A Maple egy grafikus objektumot rendel a négyzetrácson értelmezett kétváltozós függvényhez. Ez olyan szabályos Maple objektum, amelyet a szomszédos mintapontok közti lineáris interpolációra használ föl a rendszer.

```
> P; # display the surface
```



15.81. ábra: A $z = xy^2$ felület durva beosztáshoz tartozó képe

Először a **plot3d**-nek a függvényt megadó első argumentuma értékelődik ki. Ezután sorfolytonosan haladva az osztáspontoknak megfelelő függvényértékeket számítja ki a rendszer. A gyorsabb megjelenítés érdekében ezek a számítások általában a beépített (hardveres) lebegőpontos aritmetikát használják. A kétdimenziós ábrázolástól eltérően a rendszer akkor sem finomítja automatikusan a beosztást, ha az interpolációnál kapott egymáshoz csatlakozó szakaszok által bezárt szög túl nagy.

Azt mondtuk, hogy a grafikus rutinok *általában* hardveres lebegőpontos aritmetikával számolják a függvényértékeket. Természetesen ha a `Digits`-nek olyan nagy értéket adunk, melyet a hardver már nem támogat, a Maple-nek nincs más választása, mint hogy lemondjon a hardveres aritmetika használatáról. Arra is ügyelnünk kell, nehogy függvényeink definícióját olyan formában adjuk meg,

ami kizárja a hardveres aritmetika alkalmazását. Ha az ábrázolásnál komplex számokat is fölhasználunk, ez már lehetetlenné teszi a hardveres aritmetikát.

```
> evalhf( 2 + sqrt(3)*I );

Error, sqrt of a negative number
```

Ez igen tragikus hatással van a megjelenítés sebességére. Meggyőződhetünk arról, hogy a $z = \sqrt{x^2 + y^2}$ felület fölrajzolása a

```
> f := proc(x,y) sqrt(x^2+y^2) end:
```

függvénnyel körülbelül 25-ször gyorsabb, mint a

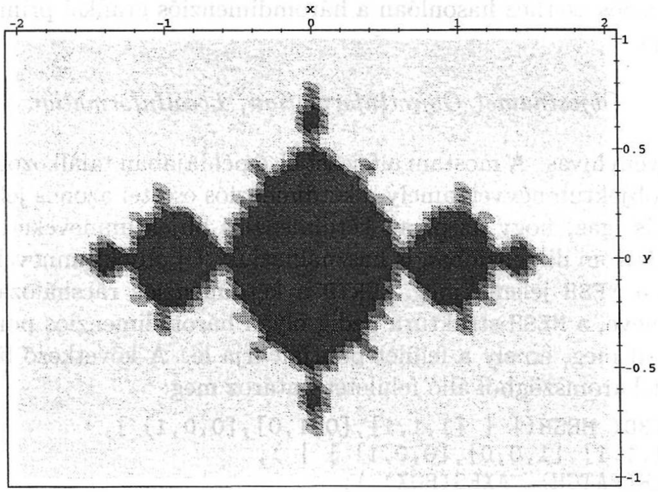
```
> g := proc(x,y) z := x+I*y; evalc( abs(z) ) end:
```

függvény esetében. Ha az Olvasó túl mesterkéltnek érezné az előbbi példát, tekintsük az $f : z \longmapsto z^2 - 1$ komplex függvény kitöltött Julia-halmazának ábrázolását (a halmaz azokból a pontokból áll, amelyekre az f iterált alkalmazásával kapott sorozat nem tart a végtelenhez). A 15.82. ábrán ezt a halmazt a „végtelenbe tartó” z_0 pontok halmazával közelítettük (a z_0 pontot „végtelenbe tartónak” tekintjük, ha az első 25 iteráció során előfordul olyan szám, amelynek abszolút értéke 3-nál nagyobb). Az ilyen pontokhoz az $F(z_0) = 1$ függvényértéket rendeljük, a többi pontra legyen $F(z_0) = 0$. A 15.82. ábra valójában az F fönti értékeit szürkeárnyalatokkal megjelenítő grafikonja fölülről nézve. A megfelelő Maple kód lehetne a következő:

```
> F := proc(x, y)
>   local z;
>   z := evalf(x + y*I ) ;
>   to 25 while abs(z)<=3 do z := z^2 -1 od:
>   if abs(z)>3 then 1 else 0 fi
> end:
> plot3d( F, -2..2, -1..1, grid=[200,200],
>   orientation=[-90,0], style=patchnogrid, axes=box,
>   labels=[x,y,' '], shading=zgrayscale );
```

Ez a változat azonban sokkal lassúbb, mint az alábbi „igazi” verzió, amely kihasználja a hardveres aritmetikát:

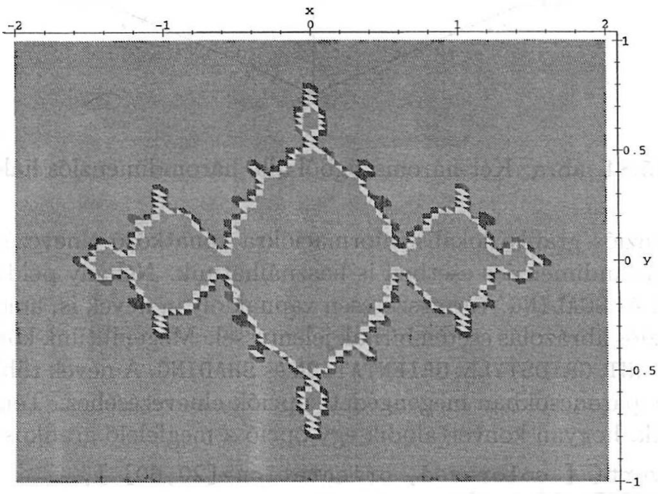
```
> F := proc(x, y)
>   local X, Y, XCOPY;
>   X := evalf(x): Y := evalf(y):
>   to 25 while X^2 + Y^2 <= 9 do
>     XCOPY := X: X := X^2-Y^2-1: Y := 2*XCOPY*Y
>   od:
>   if X^2 + Y^2 > 9 then 1 else 0 fi
> end:
> plot3d( F, -2..2, -1..1, grid=[200, 200],
>   orientation=[-90,0], style=patchnogrid, axes=box,
>   labels=[x,y,' '], shading=zgrayscale );
```



15.82. ábra: A $z \longmapsto z^2 - 1$ komplex leképezés kitöltött Julia-halmaza

A kitöltött Julia-halmaz grafikonjának határát (valójában ezt értettük Julia-halmazon) úgy kaphatjuk meg, hogy a $z = 0$ síkot ábrázoljuk a HUE színmodellben az F-et használva színfüggvényként. (Lásd 15.83. ábra.)

```
> plot3d( 0, -2..2, -1..1, grid=[75,75],
> orientation=[-90,0], style=patchnogrid,
> axes=frame, labels=[x,y,‘ ‘], color=F );
```



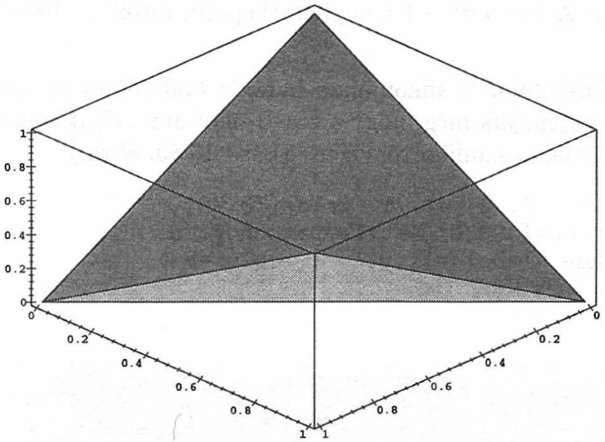
15.83. ábra: A $z \longmapsto z^2 - 1$ komplex leképezés Julia-halmaza

A kétdimenziós esethez hasonlóan a háromdimenziós grafikai primitívek általános formája

Objectname(*ObjectInformation*, *LocalInformation*)

alakú függvényhívás. A mostani alfejezet első példájában találkozott az Olvasó a `POLYGONS` objektumnévvel, amely a kétdimenziós esettel azonos jelentéssel bírt. Általában is igaz, hogy az olyan kétdimenziós objektumneveket, mint `POINT` vagy `TEXT` három dimenzióban is használhatjuk. Új objektumnevekként csupán a `GRID` és a `MESH` jelenik meg. `GRID` a kétdimenziós rácshálózatot definiáló objektum neve, a `MESH` struktúra pedig olyan háromdimenziós pontok listáiból álló listát ad meg, amely a felület pontjait írja le. A következő függvényhívás például két háromszögből álló felületet határoz meg:

```
> PLOT3D( MESH([ [ [1,1,1],[0,1,0],[0,0,1] ],
> [ [1,1,1],[1,0,0],[0,0,1] ] ] ),
> STYLE(PATCH), AXES(BOX) );
```



15.84. ábra: Két háromszögből álló háromdimenziós hálózat

A kétdimenziós grafika lokális információkra vonatkozó elnevezései közül számosat a háromdimenziós esetben is használhatunk. Néhány példa a sok közül: `AXES`, `FONT` és `SCALING`. Természetesen vannak olyan nevek is, amelyek csak háromdimenziós ábrázolás esetén bírnak jelentéssel. Megemlítünk közülük is párat: `AMBIENTLIGHT`, `GRIDSTYLE`, `ORIENTATION` és `SHADING`. A nevek többnyire hasonlóak a plot parancsokban megengedett opciók elnevezéséhez. Ténylegesen meg is nézhetjük, hogyan konvertálódik egy opció a megfelelő grafikus primitívvé:

```
> convert( [ color=red, orientation=[20,60] ],
>   PLOT3Doptions );
```

[ORIENTATION(20., 60.), COLOUR(*RGB*, 1.00000000, 0, 0)]

A plot opciók grafikus primitívekké való ilyen konverziója hasznos lehet, ha egy létező grafikus objektum valamelyik alacsony szintű grafikus primitívjét akarjuk megváltoztatni.

15.12. Speciális háromdimenziós ábrák

A fejezet korábbi részeiben főleg a kétdimenziós grafikával kapcsolatban vizsgáltuk a Maple `plots` és `plottools` csomagjait, amelyek a két- és háromdimenziós ábrázoláshoz nyújtanak segédfüggvényeket és építőelemeket. A csomagok eljárásaival speciális háromdimenziós ábrák is generálhatók. Töltsük be tehát a csomagokat, és készítsünk néhány példát:

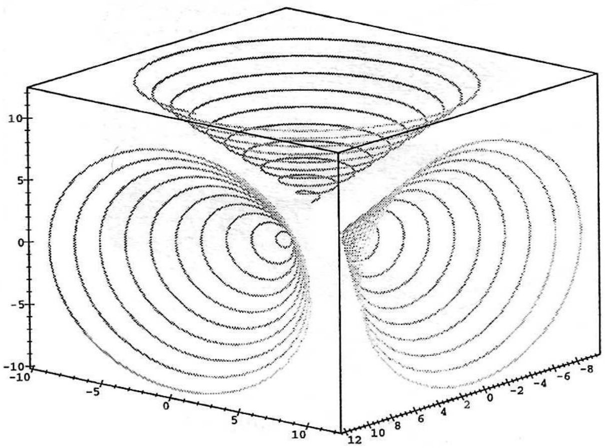
```
> with(plots): with(plottools):
```

Nézzünk először a `plots` csomag eljárásait használó példákat.

Paraméteres ábrák

Háromdimenziós térgörbéket a **spacecurve** eljárással rajzolhatunk (15.85. ábra).

```
> spacecurve( {
>    [t*cos(2*Pi*t),t*sin(2*Pi*t),2+t],
>    [2+t,t*cos(2*Pi*t),t*sin(2*Pi*t)],
>    [t*cos(2*Pi*t),2+t,t*sin(2*Pi*t)] }, t=0..10,
>   numpoints=400, orientation=[40,70],
>   style=line, thickness=3, axes=box );
```



15.85. ábra: Háromdimenziós térgörbe

A Descartes-féle koordinátarendszerben az

$$x = f(s,t),\ y = g(s,t),\ z = h(s,t)$$

paraméteres *formulákkal* definiált háromdimenziós felületet megrajzoló eljárás általános alakja

$$\textbf{plot3d}(\ [f(s,t),\ g(s,t),\ h(s,t)],\ s=a\,..\,b,\ t=c\,..\,d,\ options\ )$$

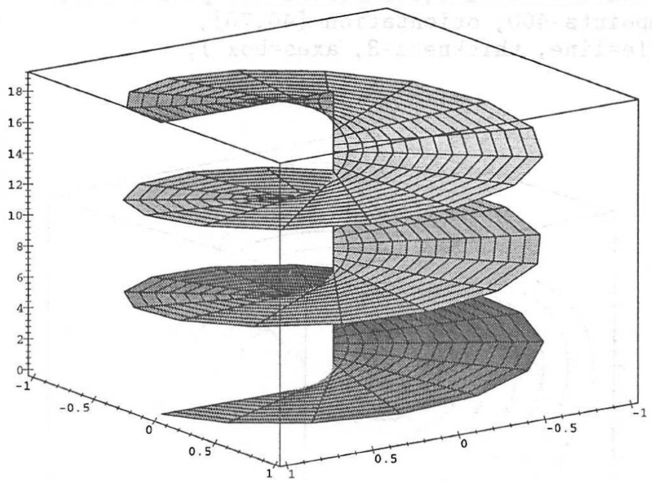
ahol $a\,..\,b$ és $c\,..\,d$ az s és a t független változók értéktartományai, *options* pedig nulla vagy több opció leírása.

Példaként ábrázoljuk a

$$(\phi, z) \longmapsto (r\cos\phi,\ r\sin\phi,\ \phi\ )$$

paraméterezés szerinti spirális felületet („helicoid") a $0 \le r \le 1$ és $0 \le \phi \le 6\pi$ értékekre.

```
> plot3d( [ r*cos(phi), r*sin(phi), phi ],
>    r=0..1, phi=0..6*Pi, grid=[15,45], style=patch,
>    orientation=[55,70], shading=zhue, axes=box );
```

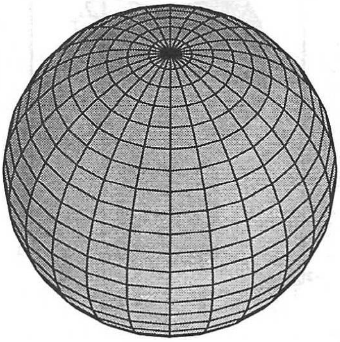


15.86. ábra: Helicoid

Koordinátarendszerek

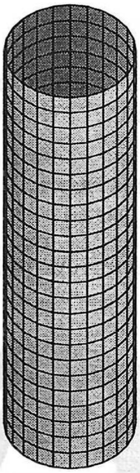
A Maple a szférikus (15.87. ábra) és a henger (15.88. ábra) mellett még sok egyéb koordinátarendszert is támogat. A következő néhány példában először elnevezzük, majd kirajzoltatjuk a grafikus objektumokat:

```
> S := plot3d( 1, theta=0..2*Pi, phi=0..Pi, style=patch,
>   coords=spherical, scaling=constrained ): S;
```



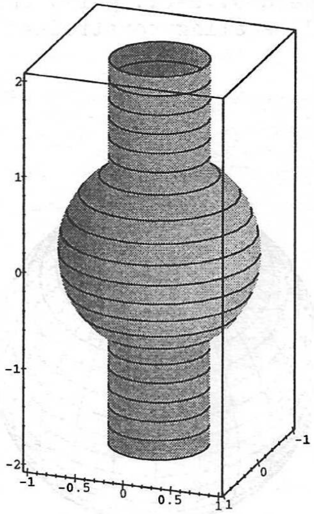
15.87. ábra: Gömb

```
> C := plot3d( 1/2, theta=0..2*Pi, z=-2..2, style=patch,
>   coords=cylindrical, scaling=constrained ): C;
```



15.88. ábra: Henger

```
> subs( STYLE(PATCH) = STYLE(PATCHCONTOUR),
>   display( {S,C}, axes=box, orientation=[20,70],
>   scaling=constrained ) );
```



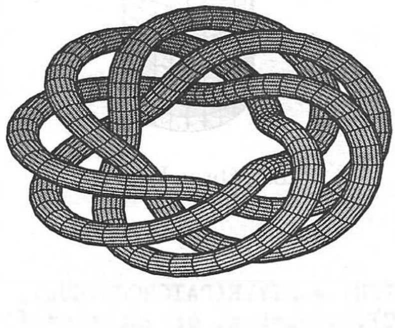
15.89. ábra: Gömb és henger kombinációja

A `style=patchcontour` opció helyett a fönti helyettesítéssel kellett a 15.89. ábrához szükségesre változtatni a megjelenítés stílusát. Ennek oka az, hogy a **display** eljárásban a szín és a stílus meghatározásához hasonló globális opciók a megfelelő grafikus objektumokhoz rendelt lokális értékekké transzformálódnak.

Csőábrák

A **tubeplot** eljárás egy vagy több háromdimenziós térgörbe körüli „csövet” definiál. Ábrázoljuk ilyen módon a $(4,7)$ típusú tóruszt. (Lásd 15.90. ábra.)

```
> r := a + b*cos(n*t):  z := c*sin(n*t):
> curve:=[ r*cos(m*t), r*sin(m*t), z ]:
> a:=2:  b:=4/5:  c:=1:  m:=4:  n:=7:
> tubeplot( curve, t=0..2*Pi, radius=1/4, numpoints=200,
>   tubepoints=20, orientation=[45,10], style=patch,
>   shading=xyz );
```

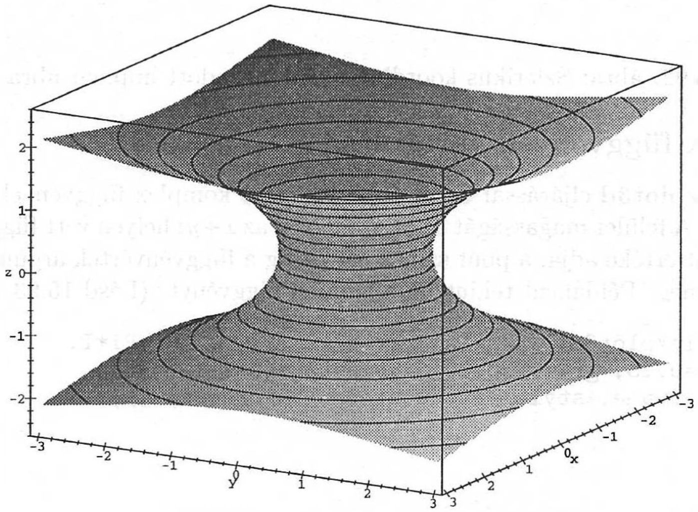


15.90. ábra: $(4,7)$ típusú „összecsomózott” tórusz

Implicit ábrázolás

Az **implicitplot3d** eljárással tetszőleges koordinátarendszerben fölírt egyenlettel definiált nemszinguláris felület megjeleníthető. A katenoid nevű felület definíciója Descartes-féle koordinátarendszerben $\cosh z = \sqrt{x^2 + y^2}$, kirajzoltatását így végezhetjük el (v. ö. 15.91. ábra):

```
> implicitplot3d( cosh(z)=sqrt(x^2+y^2),
>   x=-3..3, y=-3..3, z=-2.5..2.5, grid=[15,15,20],
>   style=patchcontour, contours=15, axes=box,
>   orientation=[30,70] );
```



15.91. ábra: A katenoid implicit ábrázolása

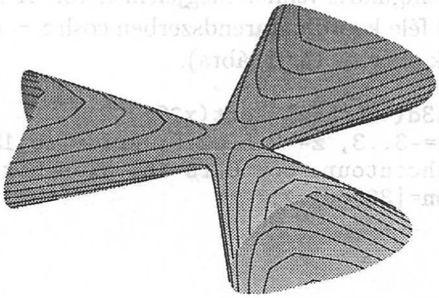
A `style=patchcontour` opció biztosítja, hogy a felületen az árnyékolás mellett a szintvonalak is láthatók legyenek.

Látványosabb implicit ábrát eredményez a szférikus koordinátákkal fölírt, szintvonalakat is tartalmazó következő példa. (Lásd 15.92. ábra.)

```
> r * ( 4 + 2*cos(3*theta) - cos(3*theta+2*phi)
> - cos(-3*theta+2*phi) + 2*cos(2*phi)) = 2;
```

$$r\,(4 + 2\cos(3\,\theta) - \cos(3\,\theta + 2\,\phi) - \cos(3\,\theta - 2\,\phi) + 2\cos(2\,\phi)) = 2$$

```
> implicitplot3d( ", r=0..2, theta=0..2*Pi, phi=0..Pi,
> coords=spherical, grid=[10,30,30],
> style=patchcontour, thickness=2, orientation=[35,30] );
```

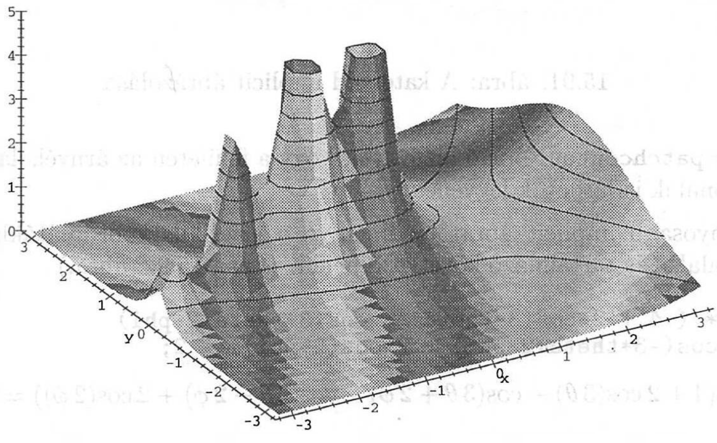


15.92. ábra: Szférikus koordinátákkal megadott implicit ábra

Komplex függvények ábrázolása

A **complexplot3d** eljárással ℂ-ből ℂ-be leképező komplex függvényeket ábrázolhatunk. A felület magasságát az (x, y) helyen az $x+yi$ helyen vett függvényérték abszolút értéke adja, a pont színezését pedig a függvényérték argumentuma határozza meg. Példaként tekintsük a gamma függvényt. (Lásd 15.93. ábra.)

```
> complexplot3d( GAMMA(z), z= -Pi-Pi*I .. Pi+Pi*I,
> view=0..5, grid=[30,30], orientation=[-120,45],
> axes=frame, style=patchcontour, thickness=2 );
```



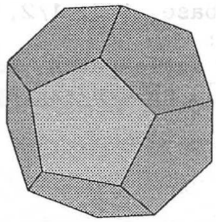
15.93. ábra: A komplex gamma függvény grafikonja

Hasonlítsuk össze ezt a rajzot a 2.6. alfejezet 2.8. ábrájával.

Poliéderek

A **polyhedraplot** eljárással könnyen ábrázolhatók az olyan ismert poliéderek, mint a tetraéder, az oktaéder vagy a dodekaéder. (Lásd 15.94. ábra.) A help fájlból való a következő példa:

```
> polyhedraplot( [0,0,0], polytype=dodecahedron,
> style=patch, scaling=constrained, orientation=[71,66] );
```



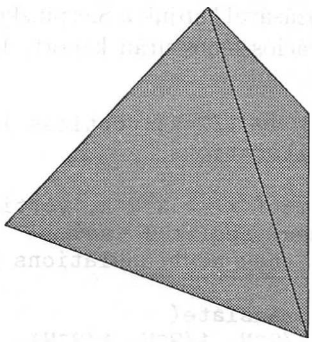
15.94. ábra: Dodekaéder

A Sierpinsky-féle tetraéder

A `plottools` csomag lehetőségeit illusztráló példáink sorát a Sierpinsky-féle tetraéder létrehozása nyitja. Ezen a példán azt is tanulmányozhatjuk, hogyan skálázhatók és tolhatók el meglévő grafikus objektumok, így a Maple-ben végezhető grafikus programozás szerény példájának tekinthető.

Először az iteráció kezdőobjektumaként szolgáló tetraédert készítjük el. Könnyen boldogulhatunk a **tetrahedron** rutinnal:

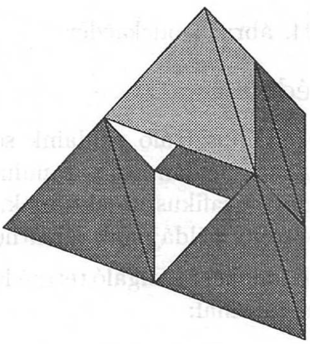
```
> base := tetrahedron( [0,0,0], 1 ):
> display( base,style=patch, scaling=constrained,
> orientation=[60,50], shading=zgrayscale );
```



15.95. ábra: A Sierpinsky-féle tetraéder konstrukciójának alapjául szolgáló standard tetraéder

Ezután $\frac{1}{2}$-szeresére kicsinyítjük, majd a kicsinyített tetraéderből négy másolatot készítünk, melyeket a négy csúcsba eltolva helyezünk el. (Lásd 15.96. ábra.)

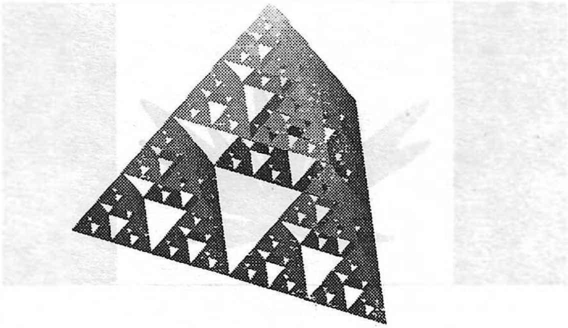
```
> vertices := {
>   [ 0, 0, sqrt(3) ],
>   [ 0, 2*sqrt(2)/sqrt(3), -1/sqrt(3) ],
>   [ -sqrt(2), -sqrt(2)/sqrt(3), -1/sqrt(3) ],
>   [ sqrt(2), -sqrt(2)/sqrt(3), -1/sqrt(3) ]}:
> translations := map( x -> 1/2*x, vertices ):
> tetrahedra := { seq(
>   translate( scale( base, 1/2, 1/2, 1/2 ), op(t) ),
>   t=translations ) }:
> display( tetrahedra, style=patch, scaling=constrained,
>   orientation=[60,50], shading=zgrayscale, thickness=2 );
```



15.96. ábra: A Sierpinsky-féle tetraéder konstrukciójának első lépése

A fönti lépések ismételt alkalmazásával kapjuk a Sierpinsky-féle tetraédert. Vessünk egy pillantást a négy iterációs lépés után kapott, 1024 tetraéderből álló alakzatra (17.97. ábra):

```
> translations := map( x -> 1/2*x, vertices ):
> N := 4:  # number of iterations
> for k from 2 to N do
>   newtranslations := map( x -> x/2^k, vertices ):
>   translations := { seq( seq( t + tnew,
>     t=translations ), tnew=newtranslations )}:
> od:
> tetrahedra := { seq( translate(
>         scale( base, 1/2^N, 1/2^N, 1/2^N),
>         op(t) ), t=translations ) }:
> display( tetrahedra, style=patchnogrid, scaling=
>   constrained, orientation=[55,50], shading=zgrayscale );
```



15.97. ábra: A négy iterációval kapott Sierpinsky-féle tetraéder

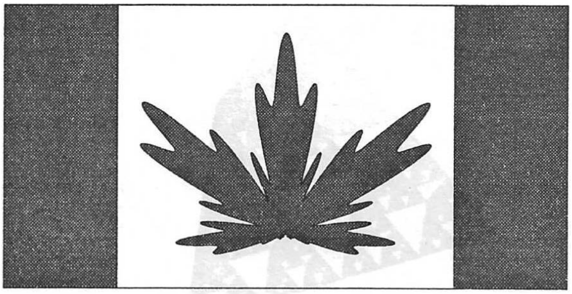
A kanadai zászló

A következő érdekes példa, melyben a kanadai zászlóhoz hasonló ábrát állítunk elő, jól illusztrálja a kétdimenziósból a háromdimenziós grafikába való transzformációkat és a grafikus objektumokon értelmezett egyéb leképezéseket. A 15.1. alfejezetben létrehozott Maple logo-ból indulunk ki:

```
> S := t -> 100/(100+(t-Pi/2)^8): # for scaling
> R := t -> S(t)*(2-sin(7*t)-cos(30*t)/2):
> mapleleaf := plot( [ R, t->t, -Pi/2..3/2*Pi],
> coords=polar, axes=none, color=red, numpoints=1000 ):
```

Ezek a parancsok juharlevélhez hasonló zárt görbét generálnak. A kanadai zászlóban a görbe belsejét pirosra kell festeni. Ez legegyszerűbben azzal érhető el, ha a CURVES objektumnevet a POLYGONS névvel helyettesítjük. Továbbá fölveszünk még két téglalapot, és bekeretezzük az ábrát, hogy megkapjuk a síkba kiterített kanadai lobogót (föltételeztük, hogy alapföltételezés szerinti háttérszínként a fehér szerepel, lásd 15.98. ábra).

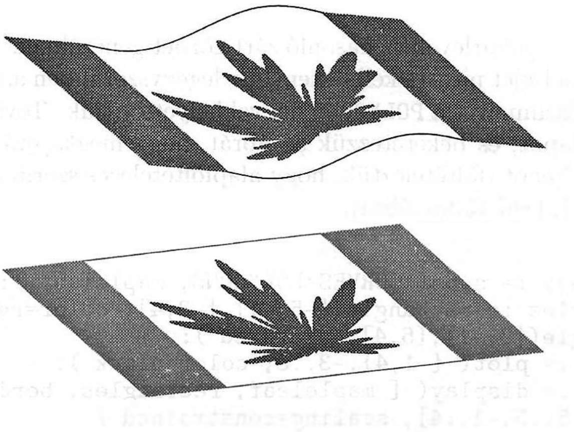
```
> mapleleaf := subs( CURVES=POLYGONS, mapleleaf ):
> rectangles := rectangle([-5,-1],[-3,4], color=red ),
> rectangle([3,-1],[5,4],color=red ):
> border := plot( {-1,4},-3..3, color=black ):
> flag2d := display( [ mapleleaf, rectangles, border ],
> view=[-5..5,-1..4], scaling=constrained ):
> flag2d;
```



15.98. ábra: A kanadai zászló hasonmása

A `plottools` csomag **transform** eljárását fölhasználva az $(x, y) \longmapsto (x, y, 0)$ transzformációval a kétdimenziós grafikus objektumot háromdimenzióssá konvertáljuk. Végül az $(x, y, z) \longmapsto (x, y, 1 + \frac{1}{15}\sin(x))$ függvénnyel a síkban kifeszített zászlót a szinuszhullámra képezzük le (15.99. ábra).

```
> flag3d := transform( (x,y,z) -> [x,y,0] )(flag2d):
> wavingflag3d :=
>   transform( (x,y) -> [x,y,1+1/15*sin(x)] )(flag3d):
> display( { flag3d, wavingflag3d },
>   scaling=unconstrained, orientation=[-110,60],axes=none,
>   style=patchnogrid, shading=none );
```



15.99. ábra: A kiterített és a hullámzó kanadai lobogó

15.13. Adatmegjelenítés

Ha függvények vagy kifejezések helyett bizonyos adatokat vagy számsorokat jelenítünk meg, speciális grafikus parancsokat kell használnunk. Megkülönböztetjük a **list**-tel kezdődő nevű *listaábrázoló parancsokat* és a `stats` csomag statisztikai rajzoló parancsait.

Listaábrázoló parancsok

A függvényeket/kifejezéseket megjelenítő némely parancshoz találhatunk vele ekvivalens listaábrázoló parancsot a `plots` csomagban. (Lásd 15.2. táblázat.)

Listaábrázolás	Megfelelője	Grafikus objektum
listplot	**plot**	adatlista 2D ábrázolása
listplot3d	**plot3d**	adatlista 3D ábrázolása
listcontplot	**contourplot**	adattömb 2D szintvonalas ábrázolása
listcontplot3d	**contourplot3d**	adattömb 3D szintvonalas ábrázolása
listdensityplot	**densityplot**	adattömb sűrűségi ábrázolása

15.2. táblázat: Listaábrázoló parancsok

A következő példák némelyikében fölhasználjuk a Holland Statisztikai Hivatalnak (URL: `http://www.cbs.nl`) az utóbbi évek holland italfogyasztására vonatkozó adatsorait.

Italfajta	Egység	1985	1990	1993	1994
sör	liter	85	91	85	89
üdítő	liter	66	86	89	93
bor	liter	15.0	14.5	16.0	16.0

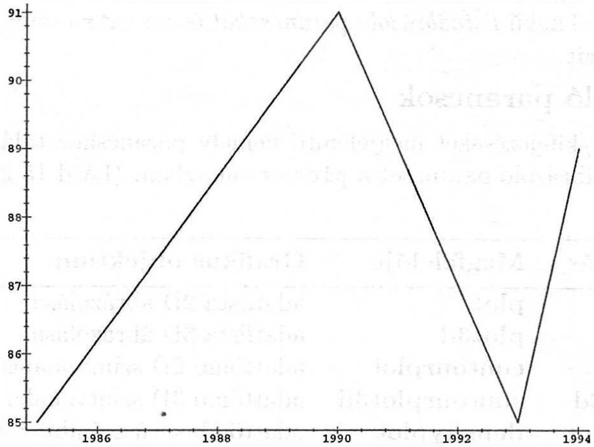
15.3. táblázat: Az egy főre eső holland italfogyasztás

A `beverage` nevű alábbi táblázat az utóbbi évek egy főre eső átlagos italfogyasztását tartalmazza listákból álló listaként:

```
> beverage[beer] :=
> [ [1985,85], [1990,91], [1993,85], [1994,89] ]:
> beverage[soft_drink] :=
> [ [1985,66], [1990,86], [1993,89], [1994,93] ]:
> beverage[wine] :=
> [ [1985,15], [1990,14.5], [1993,16], [1994,16] ]:
```

Az összes adatot a `beverage` nevű Maple táblázatban helyeztük el. Töltsük be a `plots` csomagot, és ábrázoljuk a sörfogyasztás adatait egyenes szakaszokkal összekötött pontokként.

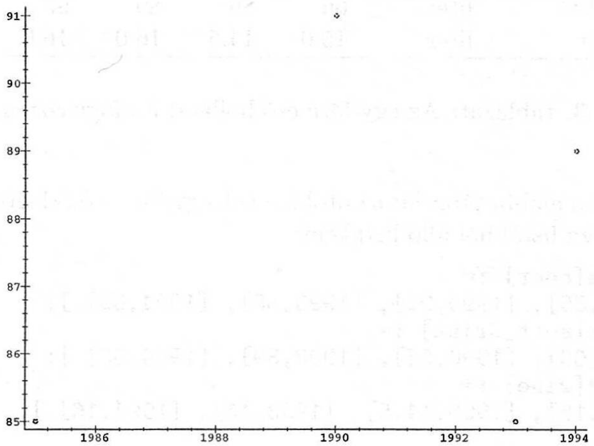
```
> with( plots ):
> listplot( beverage[beer] );
```



15.100. ábra: A holland lakosság egy főre eső éves sörfogyasztásának listás megjelenítése

A rajzolási stílus megváltoztatásával fölrajzoltathatók egyedül az adatoknak megfelelő pontok is. (Lásd 15.101. ábra.)

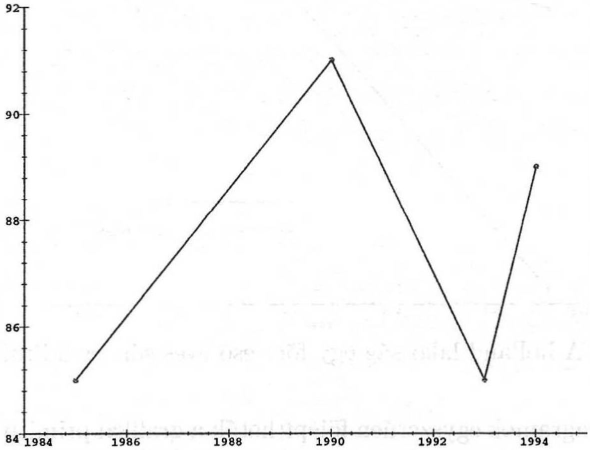
```
> listplot( beverage[beer], style=point, symbol=circle,
> color=red );
```



15.101. ábra: A holland lakosság egy főre eső éves sörfogyasztásának adatpontokból álló grafikonja

A két ábra összekombinálható és együtt is megjeleníthető a fölhasználó által definiált ábrázolási tartománnyal (15.102. ábra).

```
> display( {",""}, view=[1984..1995,84..92] );
```



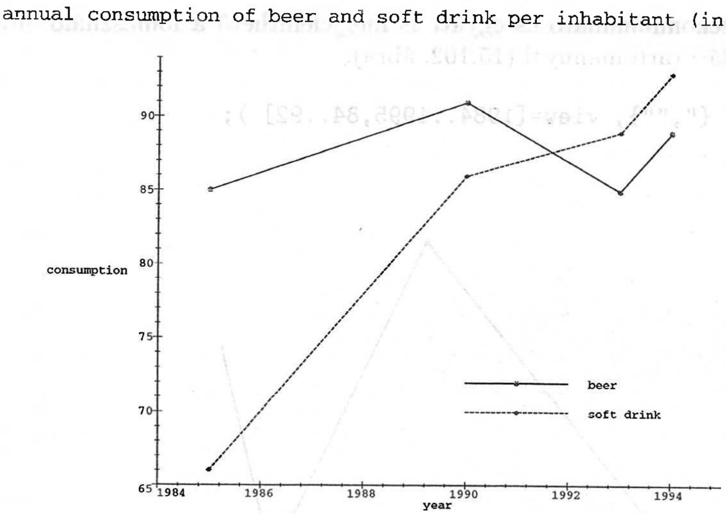
15.102. ábra: A holland lakosság egy főre eső éves sörfogyasztása

Némi erőfeszítéssel olyan ábrát is készíthetünk, amely többféle ital eltérő egyenes- és pontsstílusokkal fölrajzolt adatai mellett jelmagyarázatot is tartalmaz. (Lásd 15.103. ábra.)

```
> # Data plot:
> bp := listplot( beverage[beer], style=point,
>   symbol=circle, color=red ):
> bl := listplot( beverage[beer] ):
> sp := listplot( beverage[soft_drink], style=point,
>   symbol=diamond, color=blue ):
> sl := listplot( beverage[soft_drink], linestyle=3 ):

> # Plot legend:
> with( plottools, [line,point] ):
> bL := line( [1990,72], [1992,72] ):
> bP := point( [1991,72], symbol=circle, color=red ):
> bT := textplot( [ 1992.4, 72, beer ], align=RIGHT ):
> sL := line( [1990,70], [1992,70], linestyle=3 ):
> sP := point( [1991,70], symbol=diamond, color=blue ):
> sT := textplot( [ 1992.4, 70, `soft drink` ],
>   align=RIGHT ):

> # Complete picture:
> display({bp,bl,sp,sl,bL,bP,bT,sL,sP,sT},
>   labels=[year,consumption],
>   title=`annual consumption of beer and soft drink\
>   per inhabitant (in liter)`, view=[1984..1995,65..94] );
```



15.103. ábra: A holland lakosság egy főre eső éves sör- és üdítőfogyasztása

Oszlop- és kördiagramok egyszerűen fölépíthetők a grafikai primitívekből vagy a háromdimenziós rajzoló rutinok speciális hívásaiból. A következő részben látni fogjuk, hogyan készíthetünk oszlopdiagramokat a **histogram** statisztikai rajzoló rutinnal. Itt néhány egyéb lehetőséget mutatunk be. Először konvertáljuk adattáblánkat mátrixszá:

```
> M := map( d -> map2(op,2,op(d)), [entries(beverage)] ):
> setattribute( M, matrix );
```

$$\begin{bmatrix} 85 & 91 & 85 & 89 \\ 15 & 14.5 & 16 & 16 \\ 66 & 86 & 89 & 93 \end{bmatrix}$$

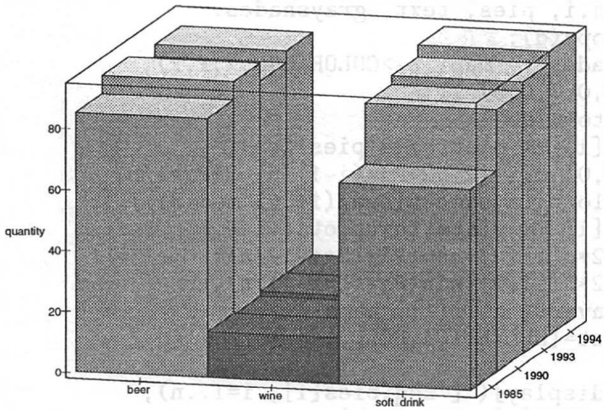
Az M mátrix sorai a következő italokat írják le:

```
> indices( beverage );
```

$$[\,beer\,], [\,wine\,], [\,soft_drink\,]$$

A `plots` csomag **matrixplot** eljárásával hisztogramként ábrázolhatjuk ezeket az adatokat (lásd 15.104. ábra)

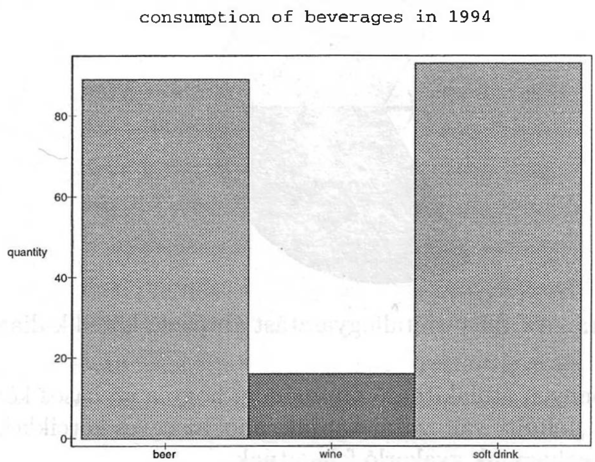
```
> matrixplot( M, heights=histogram, style=patch,
>   shading=zgrayscale, orientation=[-75,75], axes=box,
>   tickmarks=[[1.5=beer,2.5=wine,3.5=‘soft_drink‘],
>   [1.5=1985,2.5=1990,3.5=1993,4.5=1994],5],
>   labels=[“,“,quantity], font=[HELVETICA,DEFAULT,14]
> );
```



15.104. ábra: Az egy főre eső éves italfogyasztás hisztogramja

Az 1994-ben elfogyasztott italok oszlopdiagramját (lásd 15.105. ábra) így kaphatjuk meg:

```
> data94 := linalg[submatrix]( M, 1..3, 4..4 ):
> matrixplot( data94, heights=histogram, style=patch,
>   shading=zgrayscale, orientation=[-90,90], axes=box,
>   tickmarks=[[1.5=beer,2.5=wine,3.5=‘soft drink‘],0,5],
>   labels=[“,“,quantity], font=[HELVETICA,DEFAULT,14],
>   title=‘consumption of beverages in 1994‘ );
```



15.105. ábra: Az 1994-es egy főre eső holland italfogyasztás oszlopdiagramja

Kördiagramokat egyszerűen elkészíthetünk a `plottols` csomag **pieslice** eljárásával (15.106. ábra). Az alábbiakban a feladatot megoldó olyan eljárást implementálunk, amely [*italnév, mennyiség*] alakú párokból álló listákkal dolgozik:

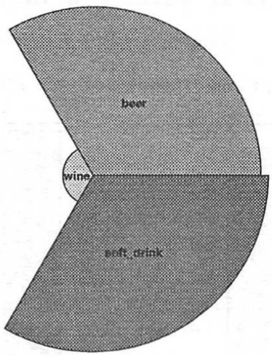
```
> circular_bar_chart := proc( d::listlist )
>   local n,i, pies, text, grayshades:
>   n := nops(d):
>   grayshades := map( x->COLOR(RGB,x,x,x),
>     [0.5,0.7,0.9] ):
>   for i to n do
>     pies[i] := plottools[pieslice](
>       [0,0],d[i][2],2*Pi*(i-1)/n..2*Pi*i/n,
>       color=grayshades[eval(1+ (i mod 3))] ):
>     text[i] := plots[textplot]( [
>       1/2*d[i][2]*cos(Pi*(2*i-1)/n),
>       1/2*d[i][2]*sin(Pi*(2*i-1)/n),
>       convert( d[i][1], name ) ],
>       font=[HELVETICA,BOLD,14] )
>   od:
>   plots[display]( [ seq(pies[i], i=1..n),
>     seq( text[i], i=1..n )], axes=none,
>     scaling=constrained )
> end:
```

Alkalmazzuk az eljárást a következő adatokra:

```
> data94 := [ seq( [ op(indices(beverage)[i]),
>   data94[i,1]], i=1..3 ) ];
```

$$data94 := [[\,beer, 89], [\,wine, 16\,], [\,soft_drink, 93\,]]$$

```
> circular_bar_chart( data94 );
```



15.106. ábra: Az 1994-es italfogyasztást ábrázoló körcikk-diagram

A fönti eljárás könnyen átalakítható oly módon, hogy a szokásos kördiagramot vagy annak olyan „fölfújt" változatát kapjuk, ahol az egyes körcikkek között kis hézag van. Ezt meghagyjuk gyakorló feladatnak.

A **pareto** paranccsal az 1994-es adatok úgynevezett *Pareto-diagramját* állíthatjuk elő. A Pareto-diagram a következő komponenseket tartalmazza: a csökkenő gyakoriságok szerint fölvett hisztogramot és az összesített gyakoriságokat jelölő görbét. Az adatok skálázásával megoldható, hogy a gyakoriságok helyett százalékértékeket használjunk.

```
> Values := map( d -> op(2,d), data94 );
```

$$Values := [89, 16, 93]$$

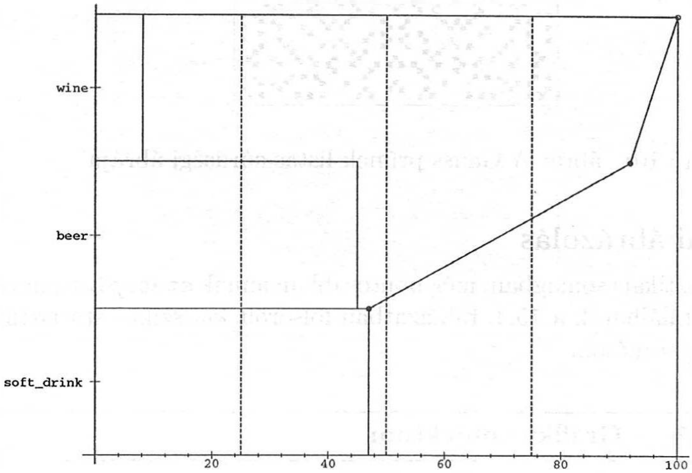
```
> Percentages := map( (x,s) -> 100*x/s,
> Values, `+`(op(Values)) );
```

$$Percentages := \left[\frac{4450}{99}, \frac{800}{99}, \frac{1550}{33}\right]$$

```
> Labels := map( d -> op(1,d), data94 );
```

$$Labels := [\,beer, wine, soft_drink\,]$$

```
> pareto( Percentages, tags=Labels );
```



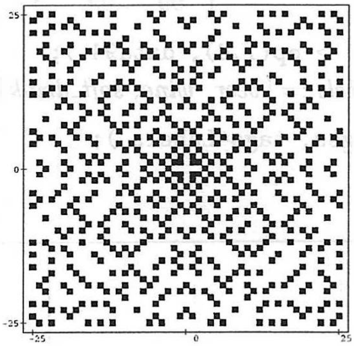
15.107. ábra: Az 1994-es italfogyasztás Pareto-diagramja

A listaábrázoló parancsokról szóló jelen szakaszt a **listdensityplot**-ra vonatkozó példával zárjuk: a komplex sík Gauss-egészeinek halmazát ábrázoljuk. Az (x, y) négyzetet feketére festjük, ha az $x + yi$ Gauss-egész prím, egyébként pedig fehéren hagyjuk. Ehhez először bevezetünk egy olyan függvényt, amely az 1 értéket veszi föl, ha a megfelelő Gauss-egész prím. A Maple-nek van egy `GaussInt` nevű csomagja a Gauss-egészekkel végzett számoláshoz, ebben található a megfelelő prímteszt is.

```
> isgaussianprime :=
>   (x,y) -> if GaussInt[GIprime](x+I*y) then
>             0
>           else
>             1
>           fi:
```

Ezután egy 50×50-es rácsot generálunk, és fölrajzoltatjuk ennek listás sűrűségi ábrázolását (15.108. ábra).

```
> data := array( [ seq( [ seq( isgaussianprime(i,j),
>   i=-25..25 ) ], j=-25..25 ) ] ):
> ticks := [ 1=-25, 26=0, 51=25 ]:
> display( listdensityplot( data, style=patchnogrid,
>   axes = box, scaling=constrained ),
>   xtickmarks=ticks, ytickmarks=ticks );
```



15.108. ábra: A Gauss prímek listás sűrűségi ábrája

Statisztikai ábrázolás

A `stats` statisztikai csomagban, még pontosabban annak `statsplots` nevű részcsomagjában találhatók a 15.4. táblázatban fölsorolt klasszikus statisztikai ábrákat készítő parancsok.

Plot utasítás	Grafikus objektum
Egydimenziós:	
boxplot	az adatokat összesítő dobozábra
histogram	az adatok hisztogramja
notchedbox	„bemetszett” dobozábra
scatter1d	pontfelhős ábra
Kétdimenziós:	
quantile	a megfigyelt értékek és kvantiliseik ábrázolása
quantile2	kvantilis-kvantilis ábrázolás
scatter2d	pontfelhős ábra
symmetry	szimmetria-ábra

15.4. táblázat: A `statsplots` csomag statisztikai ábrákat készítő parancsai

Az ábrázolás részleteinek beállítására a **changecolor**, **xshift** és **xychange** segédfüggvényeket használhatjuk. Ezen kívül a statisztikai adatokra vonatkozó görbeillesztési feladatokat is megoldhatunk a `fits` részcsomaggal. A Maple csak lineáris illesztést tud meghatározni.

Statisztikai ábrákra csak két példát hozunk. Ahogy korábban megígértük, a **histogram** eljárással is elkészítjük az 1994-es italfogyasztás oszlopdiagramját (lásd 15.109. ábra).

```
> data94;
```

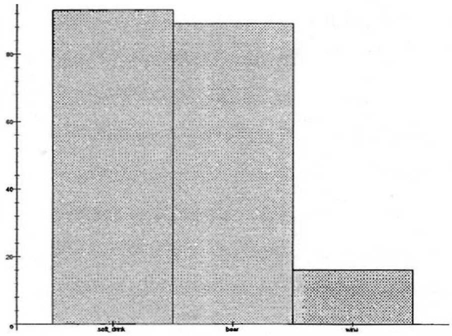
$$[[soft_drink, 93], [beer, 89], [wine, 16]]$$

A **histogram** eljárásban súlyokként használjuk ezeket az adatokat.

```
> values94 := [ seq( Weight( i..i+1, data94[i,2] ),
> i=1..3 ) ]:
```

Töltsük be a szükséges csomagokat, és rajzoltassuk föl a diagramot.

```
> with( stats ): with( statplots ):
> histogram( values94, color=yellow, tickmarks=
>   [[1.5=‘soft_drink‘, 2.5=beer, 3.5=wine], default],
>   font=[HELVETICA,DEFAULT,10] );
```



15.109. ábra: Az 1994-es italfogyasztás oszlopdiagramja

A második példában lineáris regressziót alkalmazunk numerikus adatokra, és ábrázoljuk a legkisebb négyzetek módszerével illesztett görbét. Tegyük föl, hogy a numerikus adatokat a `datafile` nevű fájlban tároljuk.

```
> ssystem(‘cat datafile‘)[2]; # display data
```

```
0       1
0.1     1.1
0.2     1.2
0.3     1.4
0.4     1.6
0.5     1.9
0.6     2.3
0.7     2.8
0.8     3.5
0.9     4.6
1       6.3
```

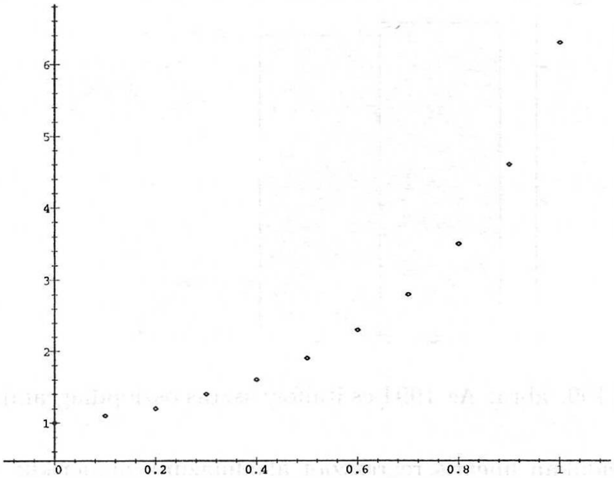
Az adatokat csökkentett numerikus pontossággal importáljuk. Beolvasásukra több módszer is van, most a `stats` csomag **importdata** parancsát fogjuk használni.

```
> with( stats): with(statplots):
> Digits := 5:
> data := importdata( datafile, 2 );
```

$$data := [0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1.], [1., 1.1, 1.2, 1.4, 1.6, 1.9, 2.3, 2.8, 3.5, 4.6, 6.3]$$

Az adatokat a 15.110. ábrán jelenítjük meg.

```
> dataplot := scatter2d( data, symbol=diamond, color=black ):
> dataplot; # display the scatter plot
```



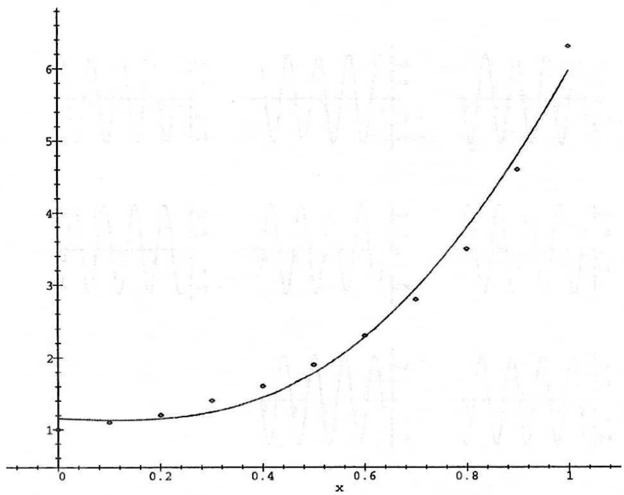
15.110. ábra: Az adatok pontfelhős ábrázolása

Ezután a legkisebb négyzetek módszerével $x \longmapsto a + b\sin(x) + c\sin(2x)$ alakú függvénnyel közelítjük a megadott értékeket, majd ábrázoljuk az eredeti adatokat és a közelítő függvényt. (Lásd 15.111. ábra.)

```
> fit[leastsquare[ [x,y], y= a + b*sin(x) + c*sin(2*x),
>   {a,b,c} ]]( [data] );
```

$$y = 1.1563 + 12.770\sin(x) - 6.5290\sin(2\,x)$$

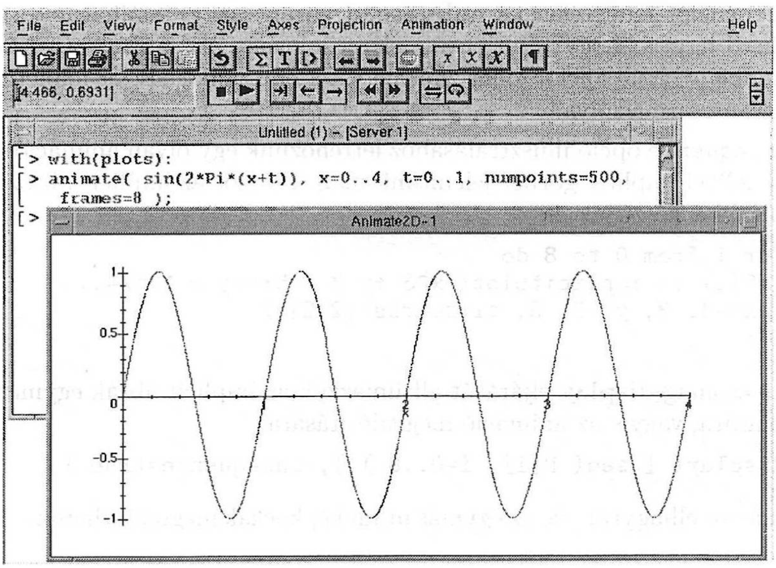
```
> curve := plot( rhs("), x=0..1, color=blue ):
> display( { dataplot, curve } );
```



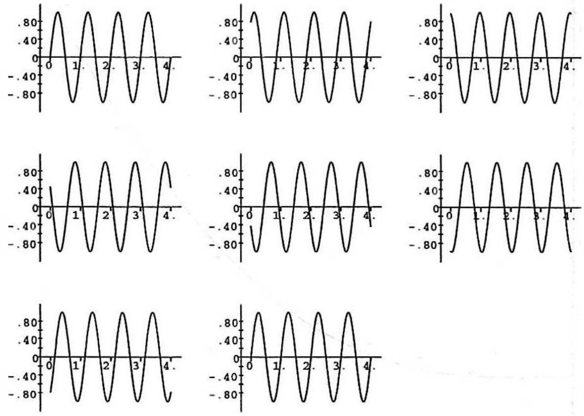
15.111. ábra: Görbeillesztés a legkisebb négyzetek módszerével

15.14. Animáció

A fejezetet néhány animációval zárjuk. A Maple animációk képkockák („frames") sorozataiból állnak, melyek gyors egymásutánban jelennek meg egy animációs ablakban. (Alapföltételezés szerint egy animáció 8 képkockából áll). A 15.112. ábrán egy mozgó szinuszhullámhoz tartozó animációs ablakot látunk. Az egyes képkockákat a 15.113. ábra tartalmazza.



15.112. ábra: Maple képernyő animációs ablakkal



15.113. ábra: A szinusz animációjának nyolc képkockája

Látható, hogy a `plots` csomag **animate** eljárásával generálhatunk animációkat. A visszajátszás módja lehet `forward`, `backward` vagy `circular`. Az animáció futtatható gyorsabban, lassabban vagy akár lépésenként. Ezeket az opciókat az ablakhoz tartozó menüsorból vezérelhetjük. Az összes többi opció azonban megadható magában az **animate** (vagy a vele analóg **animate3d** háromdimenziós) parancsban.

A 15.113. ábra 8 képkockáját valójában úgy kaptuk, hogy az animációnak megfelelő adatstruktúrát a **display** paranccsal egy grafikus tömbben rajzoltattuk ki. Formulával vagy függvénnyel generált, illetve paraméteres alakban adott háromdimenziós felületeket az **animate3d** paranccsal animálhatunk. Ugyanez elérhető úgy is, hogy több ábrát jelenítünk meg a **display** utasítással az `insequence = true` opciót fölhasználva. Kísérletezzünk kedvenc felületünk különféle transzformációival!

Az `insequence` opció illusztrálásához létrehozunk egy olyan animációs struktúrát, amellyel implicit görbét jelenítünk meg. Először az implicit ábrázolásokból álló táblát készítjük el:

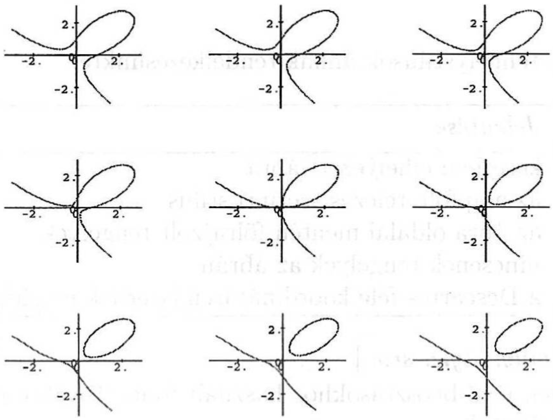
```
> for i from 0 to 8 do
>   P[i] := implicitplot( x^3 +y^3 - 5*x*y = 1-i/4,
>   x=-3..3, y=-3..3, tickmarks=[2,2] )
> od:
```

A `plots` csomag **display** eljárását alkalmazzuk az implicit ábrák egymás utáni fölrajzolására, vagyis az animáció megvalósítására:

```
> display( [ seq( P[i], i=0..8 ) ], insequence=true );
```

Az animációt elhagytuk, de az egymás utáni képkockák megtekinthetők a 15.114. ábrán.

```
> display(");
```



15.114. ábra: Algebrai görbék animációjának képkockái

Befejezésül egy olyan látványos példát ismertetünk, melyben animált oktaédert forgatunk a z tengely körül.

```
> with(plots): with(plottools): # load the plots package
> base := display( octahedron([0,0,0],1),
>   color=BLUE, scaling=constrained, axes=none,
>   style=hidden, thickness=3, orientation=[30,60] ):
> N := 16: # number of angles
> angles := [ seq( Pi*k/(2*N), k=0..(N-1) ) ]:
> for phi in angles do
>   P[phi] := rotate( base,0,0,phi)
> od:
> display( [ seq( P[phi], phi=angles ) ],
>   insequence=true );
```

Folyamatos visszajátszás esetén az animáció egyenletesen forgó oktaédert mutat.

15.15. A plotopciók listái

Ebben a részben fölsoroljuk a plot rutinok összes opcióját. A következő csoportosítást alkalmazzuk: először a minden rutinban alkalmazható opciókat írjuk le, majd az egyes grafikus módok speciális opcióit vesszük sorra. A táblázatok egy-egy sora az illető opció nevét, lehetséges értékeit és jelentését tartalmazza. Megjegyezzük, hogy az opciók értéke általában akár kis-, akár nagybetűkkel megadható, kivéve, ahol a táblázatban nagybetűs alak szerepel. Elfogadott mind a brit, mind az amerikai angol szerinti helyesírás. A szokásos alapföltételezés szerinti értékek aláhúzva szerepelnek a táblázatokban. A `DEFAULT` és a `default` speciális nevek beépített platformfüggő beállításokat jelölnek.

Általános plotopciók

`axes` = *style*

A következő tengelystílusok állnak rendelkezésünkre:

Tengelystílus	*Jelentése*
`box`	keretben elhelyezett ábra
<u>`DEFAULT`</u>	az alapföltételezés szerinti stílus
`frame`	az ábra oldalai mentén fölrajzolt tengelyek
`none`	nincsenek tengelyek az ábrán
`Cartesian`	a Descartes-féle koordinátarendszernek megfelelő ábrázolás

`axesfont` = [*family, style, size*]

A tengelyeken lévő beosztásokhoz használt font. További részletek a `font` opciónál találhatók.

`color` = *colorspec*

A színeket a következő módon adhatjuk meg:

Színspecifikáció	*Jelentése*
COLOR(RGB,*r*,*g*,*b*)	a megfelelő RGB szín
COLOR(HSV,*h*,*s*,*v*)	a megfelelő HSV szín
HUE(*h*)	COLOR(HSV,*h*,0.9,1)
color name	valamely előre definiált színnév (aquamarine, black, blue stb.) a `plot/colortable` táblából (kiíratása előtt adjuk ki a **readlib('plot/color')** parancsot)

`coords` = *name*

A megadott koordinátarendszert kívánjuk használni. Kétdimenziós grafika esetében a következő nevekre gondolhatunk: `bipolar`, `cardiod`, <u>`cartesian`</u> stb. (a **?plot,coords** kiírja a többi megengedett nevet is). A háromdimenziós grafikában gyakori koordinátarendszerek <u>`cartesian`</u>, `cylindrical`, `spherical` stb. (a **?plot3d,coords** kiírja az öszes nevet).

`font` = [*family, style, size*]

A text objektumokban használt fontot határozza meg. A *size* a pontokban megadott fontméret. A *family* és a *style* következő kombinációi érhetők el:

Fontcsalád	*Fontstílus*
COURIER	BOLD, BOLDOBLIQUE, DEFAULT, OBLIQUE
HELVETICA	BOLD, BOLDOBLIQUE, DEFAULT, OBLIQUE
TIMES	ROMAN, BOLD, ITALIC, BOLDITALIC
SYMBOL	—

`labelfont` = [*family, style, size*]

A tengelyeken elhelyezett cimkékben használt fontot határozza meg. További részleteket lásd a `font` opciónál.

`labels` = [s_x, s_y (, s_z)] (két vagy három sztring)

A tengelyek mellé írt cimkézés.

`linestyle` = n (nemnegatív egész érték)

A grafikon vonalait a 0 és 7 közti n egész értéknek megfelelő szaggatott vonalstílussal rajzoltatjuk ki. (A 0 az alapföltételezés szerinti érték.)

`numpoints` = n (természetes szám)

A mintapontok minimális számát határozza meg (az alapföltételezés szerinti értékek: két dimenzióban $n = 49$, három dimenzióban $n = 625$).

`scaling` = *mode*

Három skálázási módszert alkalmazhatunk:

Skálázás	*Jelentése*
`constrained`	az ábrát minden dimenzióban azonosan skálázzuk
`DEFAULT`	az alapföltételezés szerinti skálázás
`unconstrained`	az ábrát az egyes dimenziókban tetszőlegesen skálázhatjuk

`symbol` = *name*

Ha a `style = point` opciót alkalmazzuk, az egyes mintapontokat vagy a poligonok csúcsait a következő szimbólumok valamelyikével jelölhetjük:

Specifikáció	`box`	`circle`	`cross`	`DEFAULT`	`diamond`	`point`
Szimbólum	□	∘	+	rendszerint +	◇	·

`thickness` = n (nemnegatív egész érték)

Az ábrán szereplő egyenesek vonalvastagságát meghatározó, a [0, 15] tartományba eső szám. Ezen kívüli értékekkel a moduláris aritmetika szabályai szerint számol a rendszer. Az alapföltételezés 0, nagyobb számérték vastagabb vonalat jelent.

`tickmarks` = [*xspec*, *yspec* (, *zspec*)]

A tengelybeosztások számát, elhelyezését és cimkézését határozza meg. Mindegyik specifikáció állhat egy számból (pl. a beosztások minimális száma), számok egy listájából (pl. az osztásjelek helye a tengelyen), egyenletek listájából (pl. az osztásjelek helyei a hozzájuk tartozó cimkékkel) vagy a `default` speciális értékből.

`title` = s (sztring)

Az ábra címe (az alapértelmezés s=`NULL`, vagyis nincs cím).

`titlefont` = [*family, style, size*]

A címben használt fontot határozza meg. További részleteket lásd a `font` opciónál.

`view` = *v*

Az ábrából látható tartományokat adja meg. *v* lehet függvényértékekből álló tartomány, a `default` speciális érték, vagy ilyen specifikációkból álló egy-egy lista a tengelyirányoknak megfelelően. A `default` olyan elegendően nagy tartományt jelent, amely lehetővé teszi a teljes görbe vagy felület megjelenítését az adott dimenzióban. A `view` alapértelmezése `default` minden tengely mentén.

A plot eljárás specifikus opciói

`adaptive` = `true/false`

Az adaptív ábrázolás megengedése/tiltása.

`discont` = `true`

(*Csak kifejezésekre alkalmazható!*) Alapföltételezés szerint a Maple nem törődik a szakadási helyekkel. Ha azonban a `true` értékre állítjuk be ezt az opciót, megpróbálja megtalálni és megfelelően kezelni őket.

`filled` = `true`

Kitölti a görbe és a vízszintes tengely közti tartományt.

`resolution` = *n* (természetes szám)

A megjelenítő eszköz képpontjainak dimenziónkénti száma (alapértelmezés: $n = 200$).

`sample` = paraméterértékek sorozata

A kiindulásként használt mintapontokat megadó paraméterértékek.

`style` = *name*

A következő rajzolási stílusok állnak rendelkezésünkre:

Stílus	*Jelentése*
`line`	a mintapontokat egyenes szakaszokkal kötjük össze, a poligonok kitöltött belsejét elhagyjuk
`patch`	a pontokat szimbólumokként, a görbéket egyenes szakaszok segítségével, a poligonokat kifestett tartományokként, külön megrajzolt határukkal együtt ábrázoljuk
`patchnogrid`	ugyanaz, mint a `patch`, csak elhagyjuk a poligonok határát
`poin`	csak a mintapontokat és a poligonok csúcsait rajzolja ki

`xtickmarks` = [*xspec*]

A vízszintes tengely beosztását adja meg. Részletesebben lásd a `tickmarks` opciónál.

`ytickmarks` = [*yspec*]

A függőleges tengely beosztását adja meg. Részletesebben lásd a `tickmarks` opciónál.

A plot3d eljárás specifikus opciói

`ambientlight` = [r, g, b]

Az RGB színmodell szerint a megadott erősségű vörös, zöld és kék komponesekből álló szórt fény.

`color` = *colorfunc*

A közös plot opcióknál fölsorolt összes színmegadási lehetőségen túl ki is színezhetünk egy térgörbét vagy egy felületet. A *colorfunc* olyan függvényt vagy kifejezést jelöl, amely az ábra pontjaihoz a HUE séma szerinti színeket rendel, de lehet [r, g, b] alakú lista is, ahol r, g és b a nyomtatási tartományt meghatározó változóktól függő kifejezések vagy függvények, ekkor a kapott értékek az RGB színskála szerint értelmeződnek.

`contours` = *specification*

Ha a `contour` vagy a `patchcontour` rajzolási stílust választjuk, ezzel az opcióval megadhatjuk a szintvonalak számát. Az alapértelmezés: $n = 10$. A specifikáció lehet *függvényértékek listája* is, ekkor a nekik megfelelő szintvonalak szerepelnek az ábrán.

`grid` = [n_x, n_y] (két természetes szám)

A mintapontok által kifeszített (négyzet)rács dimenziói. Alapföltételezés: $n_x = n_y = 25$.

`gridstyle` = <u>`rectangular`</u>/`triangular`

Téglalapokból/háromszögekből álló rácsot használunk.

`light` = [ϕ, θ, r, g, b]

Az RGB színmodellnek megfelelő, a megadott erősségű vörös, zöld és kék komponesekből álló olyan fényforrást szimulál, amelynek fénye a (fokokban mért) [ϕ, θ,] szférikus koordinátákkal megadott irányból éri a grafikus objektumot.

`lightmodel` = *name*

A következő táblázat az `ambientlight` és a `light` opciókra külön-külön tartalmazza az előre definiált megvilágítási modelleket:

Modell	*Ambient Light*	*Light*
`light1`	[0.5, 0.5, 0.5]	[90, −45, 0, 0.9, 0], [45, −45, 0, 0, 0.9], [90, 45, 0.9, 0, 0]
`light2`	[0.5, 0.5, 0.5]	[45, 90, 0.9, 0, 0], [45, 45, 0, 0.9, 0], [90, −45, 0, 0, 0.9]
`light3`	[0.5, 0.5, 0.5]	[45, 45, 0, 0, 0.9], [45, 45, 0, 0.9, 0], [0, 135, 0.9, 0, 0]
`light4`	[0.6, 0.6, 0.6]	[85, 60, 0.8, 0.8, 0.8]

`orientation` = [θ, ϕ]

Azt adja meg, hogy melyik irányból nézzük az ábrát; ez a nézőpont iránya szférikus koordinátákban: ϕ a z tengellyel bezárt szög, θ pedig a nézőpont xy síkra való vetületének az x tengellyel bezárt szöge (mindkettő fokokban). Alapértelmezések: $\phi = \theta = 45$, lásd a 15.73. ábrát.

`projection` = p

A következő p értékeket használhatjuk:

p	*Projekció értéke*	*Jelentése*
`fisheye`	0	nagylátószögű perspektíva
`normal`	1/2	közepes perspektíva
`orthogonal`	1	nem perspektivikus ábra
—	$p \in [0, 1]$ valós szám	általunk definiált perspektíva

`shading` = *name*

Több árnyékolás közül választhatunk:

Árnyékolás	*Jelentése*
`none`	nem alkalmazunk árnyékolást
`xyz`	a tengelyek mentén változó színárnyalatok
`xy`	csak az x és az y tengely mentén változnak a színek
`z`	a z tengely mentén egyetlen színtartomány szerinti színezés
`zgrayscale`	a z értékek szerinti szürkeárnyalatos színezés
`zhue`	a z értékeknek megfelelő HUE színmodell szerinti színezés

`style` = *name*

A következő rajzolási stílusok használhatók:

Stílus	*Jelentése*
`contour`	csak a felület szintvonalait ábrázoljuk
`hidden`	mint a `wireframe`, csak a rejtett éleket kitöröljük, vagyis eltávolítjuk a felület más részei által takart egyenes szakaszokat
`line`	a `wireframe` szinonímája
`patch`	a felületet drótvázon kifeszített árnyékolt elemekből állítjuk össze, a takart elemek kitörlésével
`patchcontour`	szintvonalakkal kiegészített `patch` stílusú ábrázolás
`patchnogrid`	mint a `patch`, csak a rácsozatot elhagyjuk
`point`	csak a mintapontokat és a poligonok csúcsait jelenítjük meg
`wireframe`	a szomszédos mintapontokban fölvett függvényértékek egyenes szakaszokkal való összekötésével drótvázas felületet rajzolunk

Az animációval kapcsolatos opciók

`frames` = n (természetes szám)

Az animáció során n képkockát jelenítünk meg. Alapföltételezés: $n = 16$.

`framescaling` = *name*

Az animációban megjelenő képkockák skálázása a következő módokon történhet:

Skálázás	*Jelentése*
`default`	az alapföltételezés szerinti reláció
`nonuniform`	minden kocka külön skálázható
`uniform`	minden kocka azonos mérték szerint skálázott

`insequence = true` Alapföltételezés szerint a **display** több ábrát egyetlen rajzzá kombinál össze. Ezzel az opcióval az ábrákat egy animáció kockáiként jeleníthetjük meg.

15.16. Gyakorlatok

1. Ábrázoljuk a $x \longmapsto \dfrac{\sin 2x}{\sin x}$ függvényt a $(0, 4\pi)$ intervallumon.

2. Az Si szinuszintegrált a $Si(x) = \displaystyle\int_0^x \frac{\sin t}{t}\,dt$ képlettel definiáljuk. Ábrázoljuk $Si(x)$-et a $(0, 100)$ intervallumon; számítsuk ki $\lim\limits_{x\to\infty} Si(x)$-et, és az eredményt hasonlítsuk össze a a grafikonról leolvasható válasszal.

3. Ábrázoljuk az $x \longmapsto x!$ faktoriális függvényt és deriváltját a $(-4, 4)$ intervallumon.

4. Gépeljük be a

```
plot( cosh(x)^2 - sinh(x)^2 - 1, x=0..10 );
```

parancsot, és magyarázzuk meg, hogy mit látunk. Tudunk-e javítani az eredményen?

5. Ábrázoljuk az $x \longmapsto x\,\sin(1/x)$ függvényt a nulla körüli valamely „érdekes” tartományon.

6. Ábrázoljuk a $t \longmapsto \displaystyle\int_{-\infty}^t e^{-\theta^4}\,d\theta$ függvényt a $(-2, 2)$ intervallumon.

7. Ábrázoljuk az $x \longmapsto \displaystyle\int_0^x \sin(\sin\xi)\,d\xi$ függvényt a $(0, 2\pi)$ intervallumon.

8. Rajzoljunk föl néhány Lissajous-görbét, melyet tetszőleges m és n egészre a

$$t \longmapsto \big(\sin(mt), \sin(nt)\big)$$

paraméteres alak definiál. Szenteljünk külön figyelmet annak az esetnek, amikor m és n egymás utáni Fibonacci számok.

9. Rajzoljuk föl a Bernoulli-féle lemniszkátát, amely a következő módokon adható meg:

(a) az $(x^2+y^2)^2=(x^2-y^2)$ egyenlettel,

(b) az $r^2=\cos 2\phi$ polárkoordinátás egyenlettel, és

(c) az $x=\dfrac{\cos t}{1+\sin^2 t}$, $y=\dfrac{\cos t\sin t}{1+\sin^2 t}$ paraméterezéssel, melyben $-\pi\le t\le\pi$.

10. Rajzoljuk föl a következő paraméteres ábrákat.

(a) $t\longmapsto\left(\dfrac{t^2-1}{t^2+1},\dfrac{2t}{t^2+1}\right)$, $t\in(-\infty,\infty)$,

(b) $t\longmapsto(\frac{4}{9}t^3-\frac{14}{9}t^2+\frac{1}{9}t+1,-\frac{4}{9}t^3-\frac{1}{9}t^2+\frac{14}{9}t)$, $t\in(0,1)$,

(c) rajzoljuk föl az előbbi ábrát a $t\longmapsto\big(\cos(\frac{\pi}{2}t),\sin(\frac{\pi}{2}t)\big)$, $t\in(0,1)$ paraméteres függvénnyel együtt.

11. Ábrázoljuk a H Heaviside-féle lépcsősfüggvényt, majd a következő függvényeket:

(a) $H(x-1)-H(x-2)$,

(b) $\displaystyle\sum_{i=0}^{10}(-1/2)^i H(x-i/2)$,

(c) $(1-|x|)\big(H(x+1)-H(x-1)\big)$.

12. Ábrázoljuk a $t\longmapsto\big(\mathit{FresnelC}(t),\mathit{FresnelS}(t)\big)$ paraméteres alakban adott függvényt a $(0,6)$ intervallumon, és írjuk le a görbe asszimptotikus viselkedését.

13. Rajzoljuk föl az $r=\cos\theta(4\sin^2\theta-1)$ polárkoordinátás egyenlettel definiált „folium"-ot.

14. Ábrázoljuk az $x\longmapsto x+\cos(\pi x)$ függvényt a $(-49,49)$ intervallumon

(a) a **plot** alapértelmezés szerinti beállításaival,

(b) különböző opciókkal.

15. Rajzoljuk föl az $x \longmapsto e^x + \ln|4-x|$ függvényt a $(0,5)$ intervallumon. Mi a véleményünk a kapott eredményről?

16. Rajzoljuk föl az $x \longmapsto \dfrac{x}{x^2+y^2}$ függvényt -1 és 1 közti x és y értékekre.

17. Ábrázoljuk az $(x,y) \longmapsto x(x^2-3y^2)$ függvénnyel definiált „majomnyerget" különböző opciók fölhasználásával.

18. Ábrázoljuk az $(x,y) \longmapsto \sin(2\pi x)\sin(2\pi y)$ függvényt 0 és 25 közti x és y értékekre anélkül, hogy az opciók alapértelmezését megváltoztatnánk. Mi a véleményünk az eredményről?

19. Ábrázoljuk a következő paraméterezéssel adott felületet:

$$(\phi,\theta) \longmapsto (\cos\phi\sin(2\theta),\ \sin\phi\sin(2\theta),\ \sin\theta),$$

ahol θ és ϕ 0-tól 2π-ig változik.

20. Rajzoljuk föl a $(-1,0)$ és az $(1,0)$ koordinátájú pontokban elhelyezett egységnyi töltéseknek megfelelő potenciáltér szintvonalas és sűrűségi ábráját.

21. Rajzoljuk föl a Klein-féle palackot a Dieudonné-féle parametrizáció [56] fölhasználásával:

$$\begin{aligned}
x &= (2+\cos(u/2)\sin t - \sin(u/2)\sin(2t))\cos u\\
y &= (2+\cos(u/2)\sin t - \sin(u/2)\sin(2t))\sin u\\
z &= \sin(u/2)\sin t + \sin(u/2)\sin(2t)
\end{aligned}$$

ahol $0 \le u \le 2\pi$ és $0 \le t \le 2\pi$. Hasonlítsuk össze az eredményt a [128]-ban található „grafikus galéria" első példájával.

22. Rajzoljuk föl a Klein-féle palackot a Banchoff-féle parametrizációval (v. ö. [7]):

$$\begin{aligned}
x &= \big(\cos(\theta/2)(\sqrt{2}+\cos\phi) + \sin(\theta/2)(\cos\phi\sin\phi)\big)\cos\theta\\
y &= \big(\cos(\theta/2)(\sqrt{2}+\cos\phi) + \sin(\theta/2)(\cos\phi\sin\phi)\big)\sin\theta\\
z &= -\sin(\theta/2)(\sqrt{2}+\cos\phi) + \cos(\theta/2)(\cos\phi\sin\phi)
\end{aligned}$$

ahol $0 \le \theta \le 4\pi$ és $0 \le \phi \le 2\pi$. Hasonlítsuk össze az eredményt a [109]-beli színes ábrával.

23. Rajzoljuk föl a Möbius-szalagot. (Útmutatás: a megfelelő paraméterezést megkaphatjuk egyenesszakasznak változó emelkedési szögek szerinti kör körüli forgatásával.)

24. Ábrázoljuk a következő paraméterezéssel definiált katenoidot:

$$(\theta,z) \longmapsto (\cos\theta\cosh z,\ \sin\theta\cosh z,\ z),$$

a $0 \leq \theta \leq 2\pi$ és $-1 \leq z \leq 1$ paraméterértékekre. Készítsük el a megfelelő hengerkoordinátás ábrát is.

25. Ábrázoljuk azt az Enneper-féle minimális felületet, amelyet az

$$(u,v) \longmapsto \left(\frac{u}{2} - \frac{u^3}{6} + \frac{uv^2}{2}, -\frac{v}{2} + \frac{v^3}{6} - \frac{vu^2}{2}, \frac{u^2}{2} - \frac{v^2}{2}\right)$$

paraméterezéssel kapunk.

26. Ábrázoljuk az $\exp(z)\cos x = \cos y$ egyenlettel definiált Scherk-féle minimális felületet.

27. Készítsük el az $(x,y) \longmapsto \sin(xy)$ függvény szintvonalas és sűrűségi ábrázolását.

28. Ábrázoljuk az $(x,y) \longmapsto \sin x \cos y$ függvény kétdimenziós gradiensmezőjét.

29. Készítsünk forgó spirálist ábrázoló animációt.

30. Készítsük el egy ház rajzát kizárólag a grafikus primitívek fölhasználásával.

31. A szürkeárnyalatok RGB specifikációját 0 és 1 közé eső három azonos számértékkel kapjuk. Készítsük el a szürkeárnyalatok színskáláját.

32. Bizonyítsuk be és szemléltessük Pascal tételét a `geometry` csomaggal.

Pascal tétel. *Legyenek A, B, C, D, E és F egy körvonal tetszőleges pontjai. Jelölje P az* AB *és a* DF *egyenesek metszéspontját, Q a* BC *és az* FE *egyenesek metszéspontját, végül R a* CD *és az* EA *egyenesek metszéspontját. Ekkor a P, Q és az R pontok egy egyenesre illeszkednek.*

33. Bizonyítsuk be és szemléltessük Euler következő tételét.

Euler tétel. *Tetszőleges háromszögben a magasságpont (a magasságvonalak metszéspontja), a súlypont (a súlyvonalak metszéspontja) továbbá a köré írt és a beírt kör középpontja egy egyenesre illeszkedik.*

34. Igazoljuk és szemléltessük a Feuerbach-körről szóló következő tételt.

Feuerbach tétel. *A T háromszög csúcsai legyenek A, B és C. Jelölje A', B', C' az oldalfelező pontokat, D, E, F a magasságok talppontjait, végül X, Y és Z a H magasságpontot a csúcsokkal összekötő szakaszok felezőpontjait. Ekkor az A', B', C', D, E, F, X, Y és Z pontok rajta vannak azon a körön, amelynek N középpontja a H-t a T körülírt körének O centrumával összekötő szakasz felezőpontja, sugara pedig fele a T köré írható kör sugarának.*

35. A `plottools` csomaggal készítsük el a 15.106. ábrán látható kördiagram olyan „fölfújt” változatát, amelyen az egyes szegmensek között kis hézag van.

16.

Egyenletek megoldása

A fejezet a Maple-ben implementált azon módszereket tárgyalja, melyekkel különböző típusú egyenleteket és egyenlőtlenségeket, illetve ezekből álló rendszereket oldhatunk meg. Külön figyelmet szenteltünk az algebrai egyenletrendszereknek, a Gröbner-bázisok használatának. A rekurrens relációk alkotják az egyenletek másik részletesen ismertetett típusát. A (különféle tartományok fölötti) egzakt módszerek mellett foglakozunk közelítő numerikus módszerekkel is. A fejezet példái szerteágazó alkalmazási területekről származnak: elektromos áramkörök tervezése, kémiai reakciókinetika, neurális hálózatok és dimenzióanalízis.

16.1. Egyismeretlenes egyenletek

A **solve** Maple eljárás legegyszerűbb változata egyismeretlenes egyenleteket próbál megoldani:

```
> eqn := (x-1)*(x^2+x+1) = 0;
```

$$eqn := (x-1)(x^2+x+1) = 0$$

```
> solve( eqn, x );
```

$$1, -\frac{1}{2}+\frac{1}{2}I\sqrt{3}, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}$$

Az egyenlet tartalmazhat több ismeretlent is, ekkor még megkérhetjük a Maple-t, hogy oldja meg valamelyik ismeretlenre a többit paraméternek tekintve.

```
> eqn := x^3 + 2*a*x^2 + a*x = 1;
```

$$eqn := x^3 + 2\,a\,x^2 + a\,x = 1$$

```
> solve( eqn, x );
```

$$\frac{1}{6}\,\%1^{1/3} - 6\,\%2 - \frac{2}{3}\,a,$$
$$-\frac{1}{12}\,\%1^{1/3} + 3\,\%2 - \frac{2}{3}\,a + \frac{1}{2}\,I\,\sqrt{3}\,(\frac{1}{6}\,\%1^{1/3} + 6\,\%2),$$
$$-\frac{1}{12}\,\%1^{1/3} + 3\,\%2 - \frac{2}{3}\,a - \frac{1}{2}\,I\,\sqrt{3}\,(\frac{1}{6}\,\%1^{1/3} + 6\,\%2)$$
$$\%1 := 72\,a^2 + 108 - 64\,a^3 + 12\,\sqrt{-84\,a^3 - 12\,a^4 + 108\,a^2 + 81}$$
$$\%2 := \frac{\frac{1}{3}\,a - \frac{4}{9}\,a^2}{\%1^{1/3}}$$

A Maple három (komplex) megoldást talált, ezeket kifejezéssorozatként reprezentálta. A példából az is világosan kitűnik, hogyan ábrázolja a Maple a bonyolult kifejezéseket: megpróbál közös részkifejezéseket találni, és ezeket a *%*1, *%*2 stb. cimkékkel jelöli. Mi is hivatkozhatunk ezekre a cimkékre mindaddig, míg értékük meg nem változik.

```
> rationalize( %2^3 );
```

$$-\frac{1}{648}\,a^2 - \frac{1}{432} + \frac{1}{729}\,a^3 + \frac{1}{3888}\,\sqrt{-84\,a^3 - 12\,a^4 + 108\,a^2 + 81}$$

A gyakorlatban csak a közvetlen hivatkozás biztonságos, mivel előfordulhat, hogy a rendszer újra fölhasználja a cimkét, s ezzel törli az előző információt. A munkalapos fölhasználói felületen másként is kiválaszthatjuk a cimkéhez rendelt jobboldali értéket: másoljuk át a képletet egy input sorba, s ha szükséges, adjunk neki nevet is.

16.2. Rövidítési lehetőségek a **solve** alkalmazásánál

A **solve** hívásának első argumentumaként egy egyenletet vagy egyenletek halmazát várja a Maple, de ha csak kifejezéseket adunk meg, az „= 0"-t hozzá képzelve automatikusan egyenletekké egészíti ki őket.[1]

```
> solve( a + ln(x-3) - ln(x), x );
```

$$3\,\frac{e^a}{-1 + e^a}$$

[1]Fölhívjuk az Olvasó figyelmét, hogy a Maple az egyenletrendszereket és a megoldásokat *halmazokként* tárolja, így a fejezet némely példájának eredménye függ a halmazok elemeinek fölsorolásától. (*A Fordító megjegyzése.*)

Második argumentumként változónevet vagy változók halmazát kell megadnunk a Maple-nek. Ha ez hiányzik, a Maple az első argumentumban szereplő ismeretleneket az alábbihoz hasonló paranccsal határozza meg,

```
indets( eqns, name ) minus { constants }
```

és az eredményt használja a **solve** második argumentumként. Ez roppant kényelmes, de néha furcsa következményekkel járhat.

```
> solve( a + ln(x-3) - ln(x) );
```

$$\{a = -\ln(x-3) + \ln(x),\ x = x\}$$

A Maple x-re és a-ra oldotta meg a rendszert. Az $x = x$ megoldás azt jelenti, hogy x értéke tetszőleges lehet.

Láttuk, hogy a Maple a terjedelmes részkifejezéseket cimkékkel rövidíti. A rendszer néha másfajta rövidítéseket is alkalmaz:

```
> x^7 - 2*x^6 - 4*x^5 - x^3 + x^2 + 6*x + 4;
```

$$x^7 - 2\,x^6 - 4\,x^5 - x^3 + x^2 + 6\,x + 4$$

```
> solve(");
```

$$1 + \sqrt{5},\ 1 - \sqrt{5},\ \mathrm{RootOf}(_Z^5 - _Z - 1)$$

Ezzel azt akarja közölni a Maple, hogy két analitikusan fölírható megoldást talált, az $1 + \sqrt{5}$-öt és az $1 - \sqrt{5}$-öt; a többi gyök pedig a $_Z^5 - _Z - 1 = 0$ algebrai egyenletet a $_Z$ ismeretlenre megoldva kapható meg.

16.3. Néhány probléma

A **solve** alkalmazásával kapcsolatban a következő gondjaink lehetnek: a rendszer nem talál megoldást, bár biztosan tudjuk, hogy létezik, nem kapjuk meg az összes megoldást, végül túl sok (hamis) megoldást kapunk eredményül. A fönti esetek mindegyikére mutatunk példákat, és arra is teszünk javaslatokat, hogyan segíthetünk a Maple-nek.

Nem kaptunk megoldást, pedig biztosan létezik legalább egy

Gyakran tudjuk, hogy az egyenletnek vagy egyenletrendszernek van megoldása, de nincs olyan általános módszer, amellyel ezt meg is lehetne találni. Vannak olyan esetek is, amikor matematikai szempontból nézve nincs is reményünk arra, hogy analitikus formában föl tudjuk írni az általános megoldást. A Galois-elméletből jólismert, hogy az ötöd- vagy ennél magasabb fokú algebrai egyenletek gyökeire nem adható csupán gyökjelekkel fölírt megoldóképlet. A Maple-től sem várhatunk többet, csupán elég jó algoritmusok halmazát (bizonyos) egyenletek megoldására. De számos olyan matematikai probléma is létezik, amelyeknél az általános módszerek nem vezetnek eredményre, a fölhasználó mégis elő tudja

állítani valamilyen úton a megoldást. Ennek alapja sokszor a formulák speciális alakjának, tulajdonságainak fölismerése, amit egy számítógépes algebrai rendszer nem vesz észre.

Ha a Maple nem talál megoldást, három dolog történhet:

1. A Maple a **RootOf** jelöléssel visszaadja az eredeti egyenletet, hogy tovább dolgozhassunk vele:

```
> solve( cos(x) = x, x );
```

$$\mathrm{RootOf}(_Z - \cos(_Z))$$

```
> evalf(");
```

$$.7390851332$$

2. A rendszer csöndben marad, mivel nincs megoldás.

```
> solve( x = x + 1, x );
>
```

3. A rendszer csöndben marad, mivel nem talált megoldást, de elképzelhető, hogy létezik megoldás. Ezt a Maple a `_SolutionsMayBeLost` változó `true`-ra történő beállításával is megerősíti:

```
> solve( cos(x) = x^2, x );
>
> _SolutionsMayBeLost;
```

$$true$$

Beszédesebb lesz a Maple, ha megnöveljük az `infolevel[solve]` változó értékét. Mivel a **solve** is fölhasználja a `remember` opciót a korábbi problémákra adott válaszok megőrzésére, először a **forget** segítségével töröljük a **solve** előző eredményeit:

```
> readlib(forget):
> forget( solve ):  # forget previous results of solve
> infolevel[solve] := 2:  # make Maple more communicative
> solve( cos(x) = x^2, x );

solve:   Warning: no solutions found
solve:   Warning: solutions may have been lost
```

Ismert, hogy ennek az egyenletnek nincs zárt alakban fölírható megoldása. Most tehát akár meg is dicsérhetnénk a Maple-t, hogy nem talált megoldást. Néha azonban kicsit kiábrándítóak a **solve** válaszai. Az olyan számítások például, amelyek függvények valamelyik ágának kiválasztását igénylik, vagy amelyek gyököket, logaritmusokat, esetleg trigonometrikus függvényeket tartalmaznak, nem túl jól mennek a számítógépes algebrai rendszereknek, s ebből a szempontból a Maple sem kivétel.

```
> infolevel[solve] := 1:  # reset userinfo
> eqn := x + x^(1/3) = -2;
```

$$eqn := x + x^{1/3} = -2$$

```
> solve( eqn, x );
solve:   Warning: no solutions found
```

A rendszer nem talált rá az $x = -1$ valós megoldásra. Most kisegíthetjük a Maple-t, ha a gyökös kifejezéseket a **RootOf** jelöléssel algebrai számokká transzformáljuk, megoldjuk az egyenletet, az eredményt visszakonvertáljuk gyökös alakra, és ellenőrizzük, nincs-e túl sok „megoldásunk”.

```
> convert( eqn, RootOf );
```

$$x + \mathrm{RootOf}(_Z^3 - x) = -2$$

```
> solve(",x);
```

$$-1, -\frac{5}{2} + \frac{1}{2} I \sqrt{7}, -\frac{5}{2} - \frac{1}{2} I \sqrt{7}$$

Az $(x+2)^3 + x$ polinom összes komplex gyökét megkaptuk.

Másik példa:

```
> solve( sin(x) = 3*x/Pi, x );
```

$$\mathrm{RootOf}(3\,_Z - \sin(_Z)\,\pi)$$

```
> evalf(");
```

$$.5235987756$$

A $\sin(x)$ és az $x \longmapsto 3x/\pi$ függvények grafikonja alapján világosan látszik, hogy pontosan három megoldás van: $\pi/6$, $-\pi/6$ és 0.

A fentiek alapján még ne legyünk túl rossz véleménnyel a **solve** képességeiről. Néhány gyöngyszem:

```
> (x^6-1)^x = 0; solve(",x);
```

$$(x^6 - 1)^x = 0$$

$$1, \frac{1}{2} - \frac{1}{2} I \sqrt{3}, \frac{1}{2} + \frac{1}{2} I \sqrt{3}$$

```
> x + x^(1/2)+x^(1/3)+x^(1/4) = 4; solve(",x);
```

$$x + \sqrt{x} + x^{1/3} + x^{1/4} = 4$$

$$1$$

```
> x^3*(ln(5)-ln(x))=3; solve(",x);
```

$$x^3\,(\ln(5) - \ln(x)) = 3$$

$$5\,e^{(1/3\,\mathrm{LambertW}(\frac{-9}{125}))}, 5\,e^{(1/3\,\mathrm{LambertW}(-1,\,\frac{-9}{125}))}$$

```
> arccos(3*x) = 2*arcsin(x); solve(",x);
```

$$\arccos(3\,x) = 2\arcsin(x)$$

$$-\frac{3}{4} + \frac{1}{4}\sqrt{17}$$

```
> arccos(2*x) = arctan(3*x); solve(",x );
```

$$\arccos(2\,x) = \arctan(3\,x)$$

$$\frac{1}{6}\sqrt{-2 + 2\sqrt{10}}$$

```
> max( x, 2*x-2 ) = min( x^2-1, 5-x ); solve(",x );
```

$$\max(x,\,2\,x-2)=\min(x^2-1,\,5-x)$$

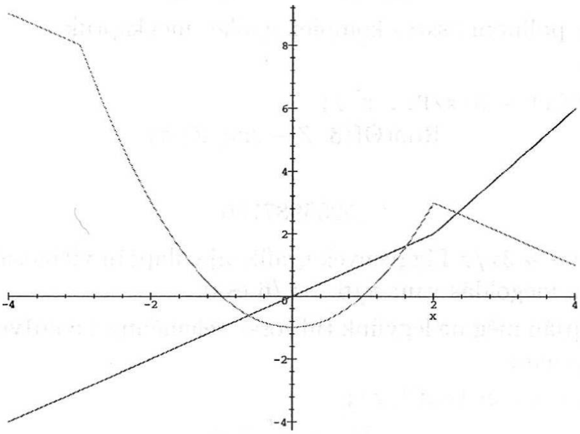
$$\frac{1}{2}-\frac{1}{2}\sqrt{5},\ \frac{1}{2}+\frac{1}{2}\sqrt{5},\ \frac{7}{3}$$

```
> evalf(["]);
```

$$[-.6180339890,\,1.618033989,\,2.333333333]$$

A 16.1. ábra alapján szemléletesen is meggyőződhetünk az eredmény helyességéről.

```
> plot( { max( x, 2*x-2 ), min( x^2-1, 5-x ) },
> x = -4..4 );
```



16.1. ábra: A $\max(x, 2x-2)$ és a $\min(x^2-1, 5-x)$ grafikonja a $(-4,4)$ intervallumon

```
> abs( x + abs(x+1) ) = 1; solve(",x );
```

$$|x+|x+1||=1$$

$$0,\ \text{RealRange}(-\infty,\ -1)$$

Az egyik megoldás (végtelen) intervallum! Ha halmazjelöléseket használunk, az eredményt is halmazokként írja ki a rendszer:

```
> solve( {""}, {x} );
```

$$\{x=0\},\ \{x\leq -1\}$$

Túl kevés megoldás

A hiányos megoldás klasszikus példája a következő:

```
> solve( sin(x) = 1/2, x );
```

$$\frac{1}{6}\pi$$

A tetszőleges $k \in \mathbb{Z}$-vel adódó végtelen sok $\frac{1}{6}\pi + 2k\pi$ és $\frac{5}{6}\pi + 2k\pi$ alakú megoldás helyett a Maple megelégszik egyetlen eggyel. A többi megoldást csak úgy kaphatjuk meg, ha az _EnvAllSolutions környezeti változónak true értéket adunk:

```
> readlib(forget):
> forget( solve ); # forget previous answer
> _EnvAllSolutions := true:
> solve( sin(x) = 1/2, x );
```

$$\frac{1}{6}\pi + \frac{2}{3}\pi_B1\tilde{} + 2\pi_Z1\tilde{}$$

A _B1~ változó tetszőleges bináris számjegyet (0 vagy 1), a _Z1~ pedig tetszőleges egészet jelöl.

```
> map( about, indets(",name) ):

Pi:
  is assumed to be: Pi

Originally _B1, renamed _B1~:
  is assumed to be: OrProp(0,1)

Originally _Z1, renamed _Z1~:
  is assumed to be: integer
```

Két további hasonló példa:

```
> solve( exp(x) = 2, x );
```

$$\ln(2) + 2 I \pi_Z\tilde{}$$

```
> solve( exp(-x) = x, x );
```

$$\mathrm{LambertW}(_NN1, 1)$$

Itt a logaritmus, illetve a Lambert-függvény különböző ágain kapjuk a különböző megoldásokat.

Néha találkozhatunk a _MaxSols változóval is, amely fölső korlátot ad meg a kért megoldások maximális számára. Alapföltételezés szerinti értéke 100, de szükség esetén ez is átállítható:

```
> eqn := product( x-k, k=1..101 ) = 0:
> nops( { solve( eqn, x ) } );
```

100

```
> _MaxSols := 200:
> nops( { solve( eqn, x ) } ); # true number of solutions
```

101

Túl sok megoldás

Egy olyan trigonometrikus egyenlet következik, amelynél a Maple támogatásra szorul.

```
> _EnvAllSolutions := true:
> eqn := sin(x)^3 -13/2*sin(x)^2 + 11*sin(x) = 4;
```

$$eqn := \sin(x)^3 - \frac{13}{2}\sin(x)^2 + 11\sin(x) = 4$$

```
> solve( eqn, x );
```

$$\arcsin(2) - 2\arcsin(2)_B1\tilde{} + 2\pi_Z1\tilde{} + \pi_B1\tilde{},$$
$$\frac{1}{6}\pi + \frac{2}{3}\pi_B1\tilde{} + 2\pi_Z1\tilde{},$$
$$\arcsin(4) - 2\arcsin(4)_B1\tilde{} + 2\pi_Z1\tilde{} + \pi_B1\tilde{}$$

Megvan az összes komplex megoldás, de mi valószínűleg csak a valósakat szerettük volna meghatározni. Jobban a dolgok mögé látunk, ha az eqn egyenletet `sin(x)`-re oldatjuk meg:

```
> solve( eqn, sin(x) );
```

$$2, \frac{1}{2}, 4$$

A kapott három egyenlet, $\sin x = 2$, $\sin x = \frac{1}{2}$ és $\sin x = 4$ már külön-külön megoldható.

A következő példa azt illusztrálja, hogy ha függvényhívások nevét használjuk ismeretlenként, s erre akarjuk megoldatni egyenleteinket, kellő óvatossággal kell eljárnunk, mert ilyenkor a Maple nem túl sok matematikai böcsességről tesz tanúbizonyságot.

```
> solve( cos(2*x) = cos(x), x );
```

$$\frac{2}{3}\pi - \frac{4}{3}\pi_B1\tilde{} + 2\pi_Z1\tilde{}, 2\pi_Z1\tilde{}$$

```
> solve( cos(2*x) = cos(x), cos(x) );
```

$$\cos(2x)$$

Vigyázat, az előző egyenletben a rendszer $\cos 2x$-et nem tekinti a $\cos x$ polinomjának.

```
> solve( 2*cos(x)^2 - 1 = cos(x), cos(x) );
```

$$\frac{-1}{2}, 1$$

```
> seq( solve( cos(x) = s, x ), s = " );
```

$$\frac{2}{3}\pi - \frac{4}{3}\pi_B1\tilde{} + 2\pi_Z1\tilde{}, 2\pi_Z1\tilde{}$$

Ezt a részt két gyakorlatiasabb kérdés vizsgálatával zárjuk. Az első az egyenletek „preprocesszálása". A gyökök megkeresése előtt a Maple habozás nélkül egyszerűsíteni próbálja az egyenletet:

```
> (x-1)^2 / (x^2-1) = 0;
```

$$\frac{(x-1)^2}{x^2-1} = 0$$

```
> solve(");
```

$$1$$

A Maple maga is rájön, hogy ha az $\frac{(x-1)^2}{x^2-1}$-et valós függvénynek tekintjük, akkor az $x = 1$ nem megoldás.

```
> subs( x=1, "" );
Error, division by zero
```

Az $\mathbb{R}[x]$ hányadostest elemeiként azonban $\frac{(x-1)^2}{x^2-1}$ és $\frac{(x-1)}{x+1}$ ekvivalensek. Az analitikus kiterjesztést véve $x = 1$ megoldásnak tekintendő.

```
> limit( lhs(""), x=1 );
```

$$0$$

Befejezésül arra a problémára hozunk föl egy példát, amikor az egyenletrendszer megoldásainak fölírásához a rendszer nem képes megfelelő paraméterezést találni.

```
> eqns := { w +x +y + z = 1, w + y = 0,
> 2*w + z = 2, v + z = 0 }:
> vars := indets( eqns );
```

$$\mathit{vars} := \{x, y, w, v, z\}$$

```
> solve( eqns, vars );
```

$$\{y = -w, v = 2w - 2, x = 2w - 1, z = -2w + 2, w = w\}$$

Ha egy egyenletrendszert az összes ismeretlenre nézve oldatunk meg, a Maple a megoldások paraméterezésére ilyesféle kritériumokat használ: „vegyük azt az egyenletet, amelyben a legkevesebb tag van" vagy „vegyük azt az egyenletet, amelyben legegyszerűbbek az együtthatók" stb. Mit tegyünk, ha más kritériumokat szeretnénk alkalmazni? Valamely változót a többiekkel kifejezve megkaphatunk úgy is, ha mellékföltételekre vonatkozó egyszerűsítéseket végeztetünk:

```
> simplify( {v,w}, eqns, [v,w,x,y,z] );
```

$$\{-z, -\frac{1}{2}z + 1\}$$

```
> simplify( {v,w}, eqns, [v,w,z,y,x] );
```

$$\{\frac{1}{2}x + \frac{1}{2}, x - 1\}$$

De ez nem azt jelenti, hogy megoldottuk az eredeti egyenletrendszert. Ez a módszer csak akkor alkalmazható, ha előre tudjuk, hogy mely változók használhatók paraméterekként. Ilyenkor a paramétereket ki is hagyhatjuk az ismeretlenek halmazából.

```
> solve( eqns, {v,w,y,z} );
```

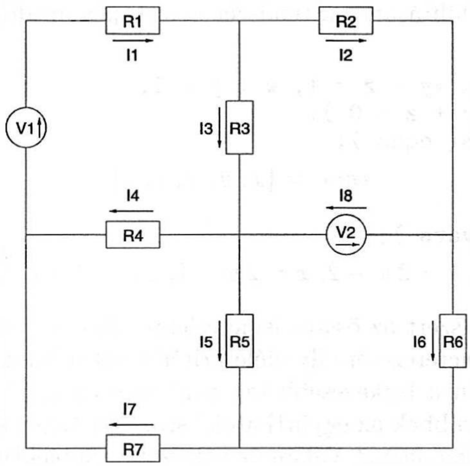
$$\{v = x - 1, w = \frac{1}{2}x + \frac{1}{2}, z = -x + 1, y = -\frac{1}{2} - \frac{1}{2}x\}$$

16.4. Egyenletrendszerek

Az előző rész utolsó példájában már láttuk, hogy a **solve** egyenletrendszerek megoldására is fölhasználható. Az egyenleteket és az ismeretleneket halmazként kell megadnunk. Még ha egyismeretlenes egyenletrendszerünk van is, az ismeretlent egyelemű halmazként kell fölírni. Ebben a részben néhány gyakorlati példán mutatjuk be az egyenletrendszerek megoldását.

Lineáris egyenletrendszerek

Először egy olyan rendszert tekintünk, amely a 16.2. ábrán látható, ellenállásokat tartalmazó elektromos áramkörben a feszültségek, az áramerősségek és az ellenállások közti kapcsolatot írja le. (Ez a példa a [62]-ből származik.)



16.2. ábra: Ellenállásokat tartalmazó elektromos áramkör

Mivel ez egy villamosmérnöki példa, célszerű az ebben a körben szokásos gyakorlatot követve a $\sqrt{-1}$-et I helyett J-vel jelölni.

```
> restart;
> alias( I=I, J=sqrt(-1) ):
> eqns := { R[1]*I[1]+R[3]*I[3]+R[4]*I[4]-V[1]=0,
>    R[2]*I[2]-V[2]-R[3]*I[3]=0, R[5]*I[5]-R[6]*I[6]-V[2]=0,
```

```
>    R[5]*I[5]+R[7]*I[7]-R[4]*I[4]=0, I[1]-I[2]-I[3]=0,
>    I[2]-I[6]-I[8]=0, I[5]+I[6]-I[7]=0, I[4]+I[7]-I[1]=0,
>    I[3]+I[8]-I[4]-I[5]=0 };
```

$$eqns := \{I_3 + I_8 - I_4 - I_5 = 0,\ I_4 + I_7 - I_1 = 0,\ I_1 - I_2 - I_3 = 0,\ I_2 - I_6 - I_8 = 0,\ I_5 + I_6 - I_7 = 0,\ R_5\, I_5 - R_6\, I_6 - V_2 = 0,\ R_1\, I_1 + R_3\, I_3 + R_4\, I_4 - V_1 = 0,\ R_2\, I_2 - V_2 - R_3\, I_3 = 0,\ R_5\, I_5 + R_7\, I_7 - R_4\, I_4 = 0\}$$

Tekintsük ezt olyan lineáris egyenletrendszernek, amelyben a feszültségek és az ellenállások adott paraméterek, az áramerősségek pedig a meghatározandó ismeretlenek.

```
> currents := { seq( I[i], i=1..8 ) }:
> resistors := { seq( R[i], i=1..7 ) }:
> voltages := { V[1], V[2] }:
> sol := solve( eqns, currents ):
```

A megoldást nem írattuk ki. Ehelyett az I_5 áramerősségre keresünk egyszerűbb képletet. Előtte még alkalmaznunk kell az **assign** eljárást az áramerősségekre kapott megoldásra, mert a **solve** magától nem teszi meg ezt.

```
> assign( sol ):
> I[5];
```

$$(R_1\, R_3\, R_7\, V_2 + R_1\, R_3\, R_4\, V_2 + R_3\, R_2\, R_4\, V_2 + R_4\, R_3\, R_7\, V_2 + R_1\, R_2\, R_4\, V_2 + R_1\, R_2\, R_7\, V_2 + R_3\, R_4\, R_6\, V_2 + R_3\, R_2\, R_7\, V_2 + V_1\, R_6\, R_2\, R_4 + V_1\, R_6\, R_4\, R_3 + R_4\, R_2\, R_7\, V_2) \Big/ (R_1\, R_3\, R_5\, R_6 + R_1\, R_3\, R_4\, R_5 + R_1\, R_3\, R_4\, R_6 + R_1\, R_3\, R_7\, R_5 + R_4\, R_3\, R_7\, R_5 + R_3\, R_2\, R_4\, R_5 + R_4\, R_3\, R_5\, R_6 + R_4\, R_3\, R_7\, R_6 + R_1\, R_3\, R_7\, R_6 + R_1\, R_2\, R_7\, R_6 + R_1\, R_2\, R_4\, R_5 + R_1\, R_2\, R_7\, R_5 + R_1\, R_2\, R_4\, R_6 + R_3\, R_2\, R_7\, R_5 + R_3\, R_2\, R_5\, R_6 + R_1\, R_2\, R_5\, R_6 + R_4\, R_2\, R_7\, R_5 + R_3\, R_2\, R_7\, R_6 + R_4\, R_2\, R_5\, R_6 + R_4\, R_2\, R_7\, R_6 + R_3\, R_2\, R_4\, R_6)$$

A formulát úgy egyszerűsítjük, hogy a részkifejezéseknek rövid neveket adunk, és I_5-öt ezekkel előállítva egyszerűsíttetünk.

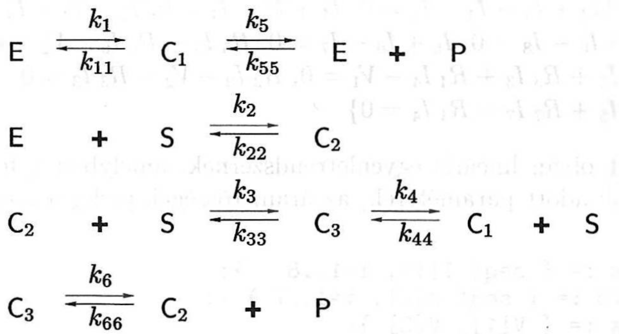
```
> relations := {
>    A = R[1]*R[2]*R[4] + R[1]*R[3]*R[4] + R[2]*R[3]*R[4],
>    B = R[5]*R[6] + R[5]*R[7] + R[6]*R[7], C = R[1]*R[2]
>        + R[1]*R[3] + R[2]*R[3] + R[2]*R[4] + R[3]*R[4],
>    D = R[4]*R[6] }:
> I[5] := map( simplify, I[5], relations,
>          [op(resistors),A,B,C,D] );
```

$$I_5 := \frac{(V_2 + V_1)\, R_3\, D + V_2\, A + V_2\, R_7\, C + V_1\, R_2\, D}{R_6\, A + B\, C + R_5\, A}$$

```
> I[5] := map( collect, I[5], [V1,V2,D] );
```

$$I_5 := \frac{((V_2 + V_1)\, R_3 + V_1\, R_2)\, D + V_2\, A + V_2\, R_7\, C}{R_6\, A + B\, C + R_5\, A}$$

Második példaként tekintsük egy olyan egyenletrendszert, amely valamely enzim által katalizált reakció pszeudo egyensúlyi állapotát írja le. A Michaelis–Menten elméletet alkalmazzuk a 16.3. ábrán látható dimerikus enzimreakcióra.



16.3. ábra: Dimerikus enzimreakció

Itt az S, P, E, C_1, C_2 és C_3 rendre az alapanyag-koncentrátumot, a végtermék koncentrátumát, a szabad enzim és a három enzim-alapanyag komplexum koncentrátumát jelenti. A reakció kinetikáját leíró determinisztikus matematikai modell a következő differenciálegyenlet-rendszerrel adható meg:

$$\begin{aligned}
S' &= k_{22}C_2 + (k_{33} + k_4)C_3 - (k_2E + k_{44}C_1 + k_3C_2)S \\
P' &= k_5C_1 + k_6C_3 - (k_{55}E + k_{66}C_2)P \\
E' &= (k_{11} + k_5)C_1 + k_{22}C_2 - (k_1 + k_2S + k_{55}P)E \\
C_1' &= (k_1 + k_{55}P)E + k_4C_3 - (k_{11} + k_5 + k_{44}S)C_1 \\
C_2' &= k_2ES + (k_6 + k_{33})C_3 - (k_{22} + k_3S + k_{66}P)C_2 \\
C_3' &= (k_{66} + P + k_3S)C_2 + k_{44}C_1S - (k_{33} + k_4 + k_6)C_3
\end{aligned}$$

Koncentráljunk az első egyenletre. A kényelem kedvéért a különféle nevekre néhány rövidítést vezetünk be a **macro** és az **alias** segítségével.

```
> alias( S=S(t), P=P(t) ): # s, p as functions of time t
> sequence := seq( k.i = k[i], i=1..6 ),
>   seq( k.i.i = k[i,i], i=1..6 ),
>   seq( C.i = C[i], i=1..6 ):
> (eval@subs)( s=sequence, 'macro(s)' ):
> RS := k22*C2 + (k33+k4)*C3 - (k2*E+k3*C2+k44*C1)*S;
```

$$RS := k_{2,2}\,C_2 + (k_{3,3} + k_4)\,C_3 - (k_2\,E + k_3\,C_2 + k_{4,4}\,C_1)\,S$$

Föltételezzük, hogy a pszeudo egyensúlyi állapotban a szabad enzim és a komplexek kontcentrációja csak olyan lassan változik, hogy akár időben állandónak is vehetjük. Ekkor a következő egyenletrendszert kapjuk, amely a C_1, C_2, C_3 és E ismeretlenekre nézve lineáris:

```
> sys :=
> { 0 = (k11+k5)*C1 + k22*C2 - (k1+k2*S+k55*P)*E,
>   0 = (k1+k55*P)*E + k4*C3 - (k11+k5+k44*S)*C1,
```

```
>    0 = k2*E*S + (k6+k33)*C3 - (k22+k3*S+k66*P)*C2,
>    0 = (k66*P+k3*S)*C2 + k44*C1*S - (k33+k4+k6)*C3,
>    E0 = C1+C2+C3+E };
```

$$
\begin{aligned}
sys := \{&0 = (k_{6,6}\,P + k_3\,S)\,C_2 + k_{4,4}\,C_1\,S - (k_{3,3} + k_4 + k_6)\,C_3,\\
&0 = (k_1 + k_{5,5}\,P)\,E + k_4\,C_3 - (k_{1,1} + k_5 + k_{4,4}\,S)\,C_1,\\
&E0 = C_1 + C_2 + C_3 + E,\\
&0 = k_2\,E\,S + (k_6 + k_{3,3})\,C_3 - (k_{2,2} + k_3\,S + k_{6,6}\,P)\,C_2,\\
&0 = (k_{1,1} + k_5)\,C_1 + k_{2,2}\,C_2 - (k_1 + k_2\,S + k_{5,5}\,P)\,E\}
\end{aligned}
$$

A Maple meg tudja oldani ezt a rendszert, így az S' reakciósebesség kifejezhető S, P, E_0 és az adott konstansok segítségével. A terjedelmes megoldást nem íratjuk ki. Ehelyett inkább számoljuk ki a $k_{55} = k_{66} = 0$-nak megfelelő speciális esetet.

```
> solve( sys, {C1,C2,C3,E} ):
> assign("):
> k55 := 0:  k66 := 0:
```

A végtermék koncentrációjára vonatkozó differenciálegyenlet a következő lesz:

```
> diff(S,t) = collect( normal(RS), S, factor );
```

$$
\begin{aligned}
\frac{\partial}{\partial t} S = -S\,E0(&k_3\,S^2\,k_2\,k_{4,4}\,k_6\\
&+ k_3\,(k_5\,k_4\,k_2 + k_{1,1}\,k_4\,k_2 + k_{1,1}\,k_6\,k_2 + k_1\,k_{4,4}\,k_6 + k_5\,k_6\,k_2)\,S\\
&- k_{4,4}\,k_1\,k_{3,3}\,k_{2,2})\Big/(k_{4,4}\,S^3\,k_3\,k_2 +\\
&(k_1\,k_{4,4}\,k_3 + k_{1,1}\,k_3\,k_2 + k_2\,k_{4,4}\,k_{3,3} + k_2\,k_{4,4}\,k_6 + k_3\,k_2\,k_4 + k_5\,k_3\,k_2)\\
&S^2 + (k_5\,k_4\,k_2 + k_1\,k_{4,4}\,k_{2,2} + k_1\,k_4\,k_3 + k_{1,1}\,k_6\,k_2 + k_{4,4}\,k_{3,3}\,k_{2,2}\\
&+ k_5\,k_4\,k_3 + k_1\,k_{4,4}\,k_6 + k_5\,k_{3,3}\,k_2 + k_{1,1}\,k_4\,k_3 + k_5\,k_6\,k_2\\
&+ k_{4,4}\,k_6\,k_{2,2} + k_{1,1}\,k_4\,k_2 + k_1\,k_{4,4}\,k_{3,3} + k_{1,1}\,k_{3,3}\,k_2)S\\
&+ k_{2,2}\,(k_{3,3} + k_4 + k_6)\,(k_1 + k_5 + k_{1,1}))
\end{aligned}
$$

A Maple ennek a differenciálegyenletnek az analitikus megoldását is meg tudja határozni a **dsolve** eljárással. A differenciálegyenletekről részletesebben csak a következő fejezetben lesz szó.

Nemlineáris egyenletrendszerek

Ennyi elég is volt a lineáris egyenletrendszerekről. Megoldásuknál egyedül a gépidő és a memória mérete jelent korlátokat. A rendszerben két algoritmust implementáltak a megoldás előállítására. Az első a ritka mátrixok kezelésével kiegészített szokásos Gauss-elimináció, a második egy „primitív" törtmentes algoritmus, amely többek között egészekből, gyökkifejezésekből vagy polinomokból álló (esetleg ritka) együtthatómátrixok esetében alkalmazható. Fordítsuk figyelmünket a nemlineáris egyenletrendszerek felé. Tekintsük a következő kör és parabola egyenletét.

$$x^2 + y^2 = 25, \quad y = x^2 - 5,$$

Oldjuk meg a rendszert a Maple segítségével.

```
> eqns := { x^2 + y^2 = 25, y = x^2 - 5 };
```

$$eqns := \{x^2 + y^2 = 25, y = x^2 - 5\}$$

```
> vars := {x,y}:
> solve( eqns, vars );
```

$$\{y = -5, x = 0\}, \{y = -5, x = 0\}, \{y = 4, x = 3\}, \{x = -3, y = 4\}$$

Ez elég egyszerűen ment, de a nemlineáris egyenletek megoldása könnyen bonyodalmakhoz vezethet:

```
> restart;
> eqns := { x^2+y^2=1, sqrt(x+y)=x^2-y^2 };
```

$$eqns := \{\sqrt{x+y} = x^2 - y^2, x^2 + y^2 = 1\}$$

```
> vars := {x,y}:
> sols := solve( eqns, vars );
```

$$sols := \{x = -\mathrm{RootOf}(2_Z^2 - 1), y = \mathrm{RootOf}(2_Z^2 - 1)\},$$
$$\{x = 1, y = 0\}, \{x = 2 - 4\,\%3^3 - 2\,\%3^2 + 3\,\%3, y = \%3\},$$
$$\{x = 2 - 4\,\%2^3 - 2\,\%2^2 + 3\,\%2, y = \%2\}$$
$$\%1 := 4_Z^4 + 4_Z^3 - 2_Z^2 - 4_Z - 1$$
$$\%2 := \mathrm{RootOf}(\%1, -.8090169944 + .3930756889\,I)$$
$$\%3 := \mathrm{RootOf}(\%1, -.3269928304)$$

A Maple láthatóan négy megoldást talált, ezeket halmazok sorozataként reprezentálta az eredmény kiírásakor. Az egyes halmazokon belül a változóknak az adott megoldáshoz tartozó értékei egyenletek formájában vannak megadva. Ez megkönnyíti a megoldások visszahelyettesítését az eredeti rendszerbe.

```
> subs( sols[2], eqns );
```

$$\{1 = 1\}$$

```
> simplify( subs( sols[1], eqns ) );
```

$$\{0 = 0, 1 = 1\}$$

A megoldások ellenőrzésére a **testeq** eljárást is használhatjuk.

```
> map( testeq, subs( sols[1], eqns ) );
```

$$\{true\}$$

Az elsőként megtalált megoldás,

```
> sols[1];
```

$$\{x = -\mathrm{RootOf}(2_Z^2 - 1), y = \mathrm{RootOf}(2_Z^2 - 1)\}$$

valójában kettő vagy négy megoldásból álló halmaz. Ha mindkét **RootOf**-os kifejezés ugyanazt a komplex számot jelöli, két megoldás van, melyeket az **allvalues** eljárással állíthatunk elő.

```
> s1 := allvalues( sols[1], 'dependent' );
```

$$s1 := \{y = \frac{1}{2}\sqrt{2}, x = -\frac{1}{2}\sqrt{2}\}, \{x = \frac{1}{2}\sqrt{2}, y = -\frac{1}{2}\sqrt{2}\}$$

Az **allvalues** hívásában a `dependent` kulcsszó azt jelenti, hogy a **RootOf** két előfordulását nem egymástól függetlenül kell kiértékelni, hanem mindkétszer ugyanazt az értéket kell venni. Az **allvalues** alkalmazásakor ez a Maple alapföltételezés szerinti viselkedése.

Ezt az első megoldást tekinthetjük négy különböző esetnek is, és külön-külön ellenőrizhetjük, hogy melyik ad igazi megoldást:

```
> sols[1];
```

$$\{x = -\mathrm{RootOf}(2_Z^2 - 1),\, y = \mathrm{RootOf}(2_Z^2 - 1)\}$$

```
> candidates := allvalues( sols[1], 'independent' );
```

$$candidates := \{y = \frac{1}{2}\sqrt{2},\, x = -\frac{1}{2}\sqrt{2}\},\, \{x = -\frac{1}{2}\sqrt{2},\, y = -\frac{1}{2}\sqrt{2}\},$$
$$\{y = \frac{1}{2}\sqrt{2},\, x = \frac{1}{2}\sqrt{2}\},\, \{x = \frac{1}{2}\sqrt{2},\, y = -\frac{1}{2}\sqrt{2}\}$$

```
> for c in candidates do
>   simplify( subs( c, eqns ) )
> od;
```

$$\{0 = 0,\, 1 = 1\}$$
$$\left\{\sqrt{-\sqrt{2}} = 0,\, 1 = 1\right\}$$
$$\{2^{1/4} = 0,\, 1 = 1\}$$
$$\{0 = 0,\, 1 = 1\}$$

Látható, hogy a négy „jelöltből” csak két esetben kaptunk megoldást. A következő eljárással válaszhatjuk ki a megoldásokat.

```
> issol := proc( sol, eqns )
>   convert( map( testeq, subs( sol, eqns ) ),
>     `and` )
> end: # the selection routine
> select( issol, {candidates}, eqns );
```

$$\left\{\left\{y = \frac{1}{2}\sqrt{2},\, x = -\frac{1}{2}\sqrt{2}\right\},\, \left\{x = \frac{1}{2}\sqrt{2},\, y = -\frac{1}{2}\sqrt{2}\right\}\right\}$$

A harmadik és a negyedik megoldásnál újabb érdekes dolgot tapasztalunk, itt ugyanis a **RootOf**-nak van egy extra argumentuma, amellyel egy meghatározott gyököt jelölünk ki. Ezt átkonvertálhatjuk a megszokottabb gyökös jelölésmódra is:

```
> sols[3];
```

$$\left\{y = \%1,\, x = 2 - 4\,\%1^3 - 2\,\%1^2 + 3\,\%1\right\}$$
$$\%1 := \mathrm{RootOf}(4_Z^4 + 4_Z^3 - 2_Z^2 - 4_Z - 1,\, -.3269928304)$$

```
> convert( ", radical );
```

$$\left\{y = \%1, x = \frac{5}{4} - 4\,\%1^3 - 2\,\%1^2 + \frac{3}{4}\sqrt{5} - \frac{3}{4}\sqrt{2+2\sqrt{5}}\right\}$$

$$\%1 := -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2+2\sqrt{5}}$$

```
> s3 := simplify(");
```

$$s3 := \left\{y = -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2+2\sqrt{5}}, x = -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2+2\sqrt{5}}\right\}$$

```
> convert( sols[4], radical );
```

$$\left\{x = \frac{5}{4} - 4\,\%1^3 - 2\,\%1^2 - \frac{3}{4}\sqrt{5} + \frac{3}{4}\sqrt{2-2\sqrt{5}}, y = \%1\right\}$$

$$\%1 := -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2-2\sqrt{5}}$$

```
> s4 := simplify(");
```

$$s4 := \left\{y = -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}I\sqrt{-2+2\sqrt{5}}, x = -\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4}I\sqrt{-2+2\sqrt{5}}\right\}$$

Ténylegesen tehát öt megoldást talált a Maple, nevezetesen

```
> s1, sols[2], s3, s4;
```

$$\left\{y = \frac{1}{2}\sqrt{2}, x = -\frac{1}{2}\sqrt{2}\right\}, \left\{x = \frac{1}{2}\sqrt{2}, y = -\frac{1}{2}\sqrt{2}\right\}, \{x = 1, y = 0\},$$

$$\left\{y = -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2+2\sqrt{5}}, x = -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2+2\sqrt{5}}\right\},$$

$$\left\{y = -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}I\sqrt{-2+2\sqrt{5}}, x = -\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4}I\sqrt{-2+2\sqrt{5}}\right\}$$

de saját jelöléseit alkalmazva a **RootOf**-okkal két vagy több gyököt összevontan, egyetlen kifejezésként írt föl. Némileg csalódott lehet az Olvasó, hogy a Maple miért nem a gyökös alakban fölírt megoldásokkal állt elő azonnal. Végül is a **RootOf**-ok belsejében legföljebb negyedfokú polinomok szerepeltek, ezek gyökeit meg ki lehet fejezni gyökjelekkel. Ha az `_EnvExplicit` környezeti változót a `true` értékre állítjuk, a Maple mindig fönti óhajunk szerint jár el.

```
> readlib(forget):
> forget( solve ):  # forget previous results
> _EnvExplicit := true:
> candidates := { solve( eqns ) };
```

$$candidates := \left\{ \{x = 1, y = 0\}, \left\{y = \frac{1}{2}\sqrt{2}, x = -\frac{1}{2}\sqrt{2}\right\},\right.$$

$$\left\{x = \frac{1}{2}\sqrt{2}, y = -\frac{1}{2}\sqrt{2}\right\},$$

$$\left\{y = \%4, x = \frac{5}{4} - 4\,\%4^3 - 2\,\%4^2 + \frac{3}{4}\sqrt{5} - \frac{3}{4}\sqrt{2+2\sqrt{5}}\right\},$$
$$\left\{y = \%3, x = \frac{5}{4} - 4\,\%3^3 - 2\,\%3^2 + \frac{3}{4}\sqrt{5} + \frac{3}{4}\sqrt{2+2\sqrt{5}}\right\},$$
$$\left\{y = \%2, x = \frac{5}{4} - 4\,\%2^3 - 2\,\%2^2 - \frac{3}{4}\sqrt{5} + \frac{3}{4}\sqrt{2-2\sqrt{5}}\right\},$$
$$\left\{y = \%1, x = \frac{5}{4} - 4\,\%1^3 - 2\,\%1^2 - \frac{3}{4}\sqrt{5} - \frac{3}{4}\sqrt{2-2\sqrt{5}}\right\}\Bigg\}$$
$$\%1 := -\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2-2\sqrt{5}}$$
$$\%2 := -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2-2\sqrt{5}}$$
$$\%3 := -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2+2\sqrt{5}}$$
$$\%4 := -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2+2\sqrt{5}}$$

```
> nops(");
```

$$7$$

Ez még túl sok megoldás! A problémát az okozta, hogy a Maple nem különböztette meg, hogy a $4z^4 + 4z^3 - 2z^2 - 4z - 1$ polinom négy gyöke közül melyekkel kaptunk valódi megoldásokat. Ezért nekünk kell kiválogatni az igazi megoldásokat. Ez nem megy a korábban definiált **issol** rutinnal, mivel az teljes egészében a **testeq** eljárásra hagyatkozik, és nem tartalmaz óvintézkedéseket arra az esetre nézve, ha az utóbbi hibázna. Ezért most úgy alakítjuk át, hogy először az összes tagot áthozzuk a baloldalra, négyzetre emelünk, majd a **radnormal** eljárással normálalakra hozzuk a skatulyázott gyökjeleket tartalmazó kifejezéseket. A feladat további része már megoldható a **testeq**-val. Ha a **testeq** hibát jelez, vagyis nem tudja eldönteni, hogy az adott kifejezés tényleg megoldás-e, a biztonság kedvéért visszadjuk ezt a gyököt is.

```
> issol := proc( sol, eqns )
>   local eqs;
>   eqs := subs( sol, eqns );
>   eqs := map( lhs - rhs, eqs );
>   eqs := map( u -> u^2, eqs ); # <- square expressions
>   eqs := radnormal( eqs );
>   if not convert( map( testeq, eqs ), `and` )
>   then false
>   else true
>   fi
> end:  # the selection routine
```

Az előző szelekciós rutinban a részeredményeket azért emeltük négyzetre, hogy lehetőség szerint elkerüljük az egymásba skatulyázott gyökjelekkel kapcsolatos problémákat. Enélkül a **testeq** nem biztos, hogy rájönne arra, hogy az $1/2\sqrt{2-2\sqrt{5}} + 1/2\sqrt{5}\sqrt{2-2\sqrt{5}}$ nemnulla komplex szám.

```
> select( issol, candidates, eqns );
```

$$\left\{ \{x=1, y=0\}, \left\{y=\frac{1}{2}\sqrt{2}, x=-\frac{1}{2}\sqrt{2}\right\}, \left\{x=\frac{1}{2}\sqrt{2}, y=-\frac{1}{2}\sqrt{2}\right\},\right.$$
$$\left\{y=\%2, x=\frac{5}{4}-4\,\%2^3-2\,\%2^2+\frac{3}{4}\sqrt{5}-\frac{3}{4}\sqrt{2+2\sqrt{5}}\right\},$$
$$\left.\left\{y=\%1, x=\frac{5}{4}-4\,\%1^3-2\,\%1^2-\frac{3}{4}\sqrt{5}+\frac{3}{4}\sqrt{2-2\sqrt{5}}\right\}\right\}$$
$$\%1 := -\frac{1}{4}-\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{2-2\sqrt{5}}$$
$$\%2 := -\frac{1}{4}+\frac{1}{4}\sqrt{5}-\frac{1}{4}\sqrt{2+2\sqrt{5}}$$

```
> nops(");
```

$$5$$

A négyzetreemelésekkel az eredeti egyenletrendszer polinomiális rendszerré alakítható. Az így nyert rendszer a hagyományos módszerekkel, például az ismeretlenek kiküszöbölésével is megoldható.

Először végezzük el a négyzetreemelést, és írjuk föl a kapott polinomokkal az új rendszert:

```
> P[1] := ( lhs - rhs )( eqns[1] );
```

$$P_1 := x^2+y^2-1$$

```
> P[2] := ( lhs^2 - rhs^2 )( eqns[2] );
```

$$P_2 := x+y-(x^2-y^2)^2$$

```
> Peqns := { P[1], P[2] };
```

$$Peqns := \{x+y-(x^2-y^2)^2, x^2+y^2-1\}$$

A következő lépésben küszöböljük ki az y ismeretlent a rá vonatkozó rezultáns kiszámításával. (A rezultáns definícióját megtaláljuk kedvenc algebra tankönyvünkben, vagy például az [53, 76, 130, 132, 138] irodalmakban.)

```
> eqn_x := resultant( P[1], P[2], y ) = 0;
```

$$eqn_x := -6\,x^2-2\,x-8\,x^5+8\,x^3+24\,x^4+16\,x^8-32\,x^6=0$$

```
> eqn_x := factor( eqn_x );
```

$$eqn_x := 2\,x\,(x-1)\,(2\,x^2-1)\,(4\,x^4+4\,x^3-2\,x^2-4\,x-1)=0$$

A Maple könnyedén megoldja az x ismeretlenre ezt az algebrai egyenletet:

```
> x_roots := solve( eqn_x );
```

$$x_roots := 0, 1, \frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{2}, -\frac{1}{4}+\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{2+2\sqrt{5}},$$
$$-\frac{1}{4}+\frac{1}{4}\sqrt{5}-\frac{1}{4}\sqrt{2+2\sqrt{5}}, -\frac{1}{4}-\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{2-2\sqrt{5}},$$
$$-\frac{1}{4}-\frac{1}{4}\sqrt{5}-\frac{1}{4}\sqrt{2-2\sqrt{5}}$$

Az y ismeretlent mellékföltételekkel kiegészített egyszerűsítéssel fejezhetjük ki x segítségével:

```
> simplify( P[2], { P[1] }, [ y, x ] );
```

$$x + y - 4x^4 + 4x^2 - 1$$

```
> eqn_y := y = solve(",y);
```

$$eqn_y := y = -x + 4x^4 - 4x^2 + 1$$

Az $\{eqn_x, eqn_y\}$ egyenletekből álló új rendszerre ugyanazokat a megoldásokat kaptuk, mint amit az `_EnvExplicit` környezeti változó `true` értékénél a **solve** adott volna. A „hamis gyökök" a négyzetreemelés következményei. Mindenesetre megtaláltuk az eredeti rendszer összes megoldását.

A kiküszöbölést könnyebb lett volna az **eliminate** könyvtári függvénnyel elvégezni.

```
> readlib( eliminate ):
> eliminate( {P[1],P[2]}, y );
```

$$[\{y = 2x^4 - x - 2x^2 + 1\},$$
$$\{x(x-1)(2x^2-1)(4x^4 + 4x^3 - 2x^2 - 4x - 1)\}]$$

Ez az eljárás pszeudo-rezultánsokat és pszeudo-szubrezultánsokat használ (v. ö. [4, 53, 76, 139]). Az **eliminate** eredménye alapján könnyen láthatók a megoldások.

Vegyük elő újra eredeti példánkat, de most írjuk föl három ismeretlennel. A négyzetgyök helyett vezessünk be egy új z ismeretlent, s konstruáljunk ezzel egy olyan algebrai egyenletrendszert, amelynek megoldáshalmaza tartalmazza az eredeti rendszer összes megoldását:

```
> eqns := { x^2 + y^2 = 1, z = x^2 - y^2, z^2 = x + y };
```

$$eqns := \{z = x^2 - y^2, z^2 = x + y, x^2 + y^2 = 1\}$$

Ezután tekintsük a rendszert definiáló polinomok halmazát:

```
> polys := map( lhs - rhs, eqns );
```

$$polys := \{z - x^2 + y^2, z^2 - x - y, x^2 + y^2 - 1\}$$

Most küszöböljük ki y-t és z-t:

```
> eliminate( polys, {y,z} );
```

$$[\{z = 4x^2 - 4x^8 + 4x^5 + 8x^6 - 8x^4 - 4x^3 + 2x - 1,$$
$$y = 2x^4 - x - 2x^2 + 1\},$$
$$\{x(x-1)(2x^2-1)(4x^4 + 4x^3 - 2x^2 - 4x - 1)\}]$$

Lényegében megkaptuk a korábban látott megoldást.

Általában nem ilyen könnyen megy az algebrai egyenletrendszerek analitikus megoldásának meghatározása az eliminációs módszerrel. A következő részben egy jóval kifinomultabb módszert írunk le ugyanennek a problémának megoldására.

16.5. A Gröbner-bázis módszer

A polinomok kiszámíthatósági elméletének fontos fogalmát alkotják a Gröbner-bázisok. A 14.7. alfejezetben a mellékföltételekre vonatkozó egyszerűsítésnél alkalmaztuk a Gröbner-bázis módszert. Ebben a részben azt fogjuk röviden megvizsgálni, hogyan használható algebrai egyenletrendszerek megoldására. További részletek (például a megoldáshalmaz végességének tesztelése vagy a módszer kiterjesztése testekről gyűrűkre) a [13, 26, 27, 51] munkákban találhatók.

Lássuk, hogyan kezelhető az előző rész utolsó példája a Gröbner-bázis módszerrel:

```
> polys := { x^2 + y^2 - 1, x^2 - y^2 - z, x + y - z^2 };
```

$$polys := \{x^2 + y^2 - 1,\ x^2 - y^2 - z,\ x + y - z^2\}$$

Az x, y és z változóknak a $z \succ y \succ x$ relációval indukált valódi lexikografikus rendezésére vonatkozó minimális Gröbner-bázist így kapjuk:

```
> with( grobner ): # load the Groebner basis package
> G := gbasis( polys, [z,y,x], 'plex' );
```

$$G := [z - 2\,x^2 + 1,\ x + y - 1 + 4\,x^2 - 4\,x^4,\\ -3\,x^2 - x + 4\,x^3 - 4\,x^5 + 12\,x^4 - 16\,x^6 + 8\,x^8]$$

```
> G := factor( G );
```

$$G := [z - 2\,x^2 + 1,\ x + y - 1 + 4\,x^2 - 4\,x^4,\\ x\,(x-1)\,(2\,x^2-1)\,(4\,x^4 + 4\,x^3 - 2\,x^2 - 4\,x - 1)]$$

A Gröbner bázis közös zérushelyeinek halmaza ekvivalens az eredeti polinomok közös zérushelyeinek halmazával. A Gröbner bázisban viszont sikerült leválasztanunk az x egyváltozós polinomját, a másik két ismeretlen pedig könnyen kifejezhető x segítségével, mivel az első két polinom triviálisan megoldható y-ra, illetve z-re.

$$z = 2x^2 - 1, \quad y = -4x^4 - 4x^2 - x + 1$$

A harmadik polinom,

$$x\,(x-1)\,(2\,x^2-1)\,(4\,x^4 + 4\,x^3 - 2\,x^2 - 4\,x - 1)$$

gyökei fölírhatók analitikus alakban. Ezután minden gyöke visszahelyettesíthető az y-ra és a z-re vonatkozó egyenletbe.

```
> _EnvExplicit := true:
> assign( { solve(G[1],{z}), solve(G[2],{y}) } );
> rootlist := [ solve( G[3] ) ];
```

$$rootlist := \left[0,\ 1,\ \frac{1}{2}\sqrt{2},\ -\frac{1}{2}\sqrt{2},\ -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\right.\\ -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\ -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2 - 2\sqrt{5}},\\ \left.-\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2 - 2\sqrt{5}}\right]$$

```
> for x in rootlist do
>   simplify( [ 'x'=x, 'y'=y, 'z'=z ] );
>     if simplify( subs( ", polys ) ) = {0}
>     then print( `valid solution` )
>     else print( `INvalid solution` )
>     fi
> od;
```

$$[\,x = 0,\, y = 1,\, z = -1\,]$$

valid solution

$$[\,x = 1,\, y = 0,\, z = 1\,]$$

valid solution

$$\left[x = \frac{1}{2}\sqrt{2},\, y = -\frac{1}{2}\sqrt{2},\, z = 0\right]$$

valid solution

$$\left[x = -\frac{1}{2}\sqrt{2},\, y = \frac{1}{2}\sqrt{2},\, z = 0\right]$$

valid solution

$$\left[x = -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\, y = -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\right.$$
$$\left. z = -\frac{1}{4}\sqrt{2 + 2\sqrt{5}} + \frac{1}{4}\sqrt{5}\sqrt{2 + 2\sqrt{5}}\right]$$

valid solution

$$\left[x = -\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\, y = -\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}\sqrt{2 + 2\sqrt{5}},\right.$$
$$\left. z = \frac{1}{4}\sqrt{2 + 2\sqrt{5}} - \frac{1}{4}\sqrt{5}\sqrt{2 + 2\sqrt{5}}\right]$$

valid solution

$$\left[x = -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4} I \sqrt{-2 + 2\sqrt{5}},\, y = -\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4} I \sqrt{-2 + 2\sqrt{5}},\right.$$
$$\left. z = -\frac{1}{4} I \sqrt{-2 + 2\sqrt{5}} - \frac{1}{4} I \sqrt{5}\sqrt{-2 + 2\sqrt{5}}\right]$$

valid solution

$$\left[x = -\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4} I \sqrt{-2 + 2\sqrt{5}},\, y = -\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4} I \sqrt{-2 + 2\sqrt{5}},\right.$$
$$\left. z = \frac{1}{4} I \sqrt{-2 + 2\sqrt{5}} + \frac{1}{4} I \sqrt{5}\sqrt{-2 + 2\sqrt{5}}\right]$$

valid solution

A Gröbner-bázis módszer egyik előnye az előző részben látott eliminációs módszerrel szemben az, hogy különböző rendezési relációkkal kísérletezve mélyebben megismerhetjük a megoldandó probléma struktúráját.

Második példánk egy neurális hálózatokkal kapcsolatos közönséges differenciálegyenlet stabil állapotait leíró egyenletrendszer (lásd [146]).

$$\{cx + xy^2 + xz^2 = 1, \quad cy + yx^2 + yz^2 = 1, \quad cz + zx^2 + zy^2 = 1\}$$

Ezt az x, y és z ismeretlenekre vonatkozó egyenletrendszernek tekintjük; c adott paraméter. A **solve** eljárással kapott alábbi megoldás algebrai függvényeket tartalmaz:

```
> eqns := { c*x + x*y^2 + x*z^2 = 1,
> c*y + y*x^2 + y*z^2 = 1, c*z + z*x^2 + z*y^2 = 1 };
```

$$eqns := \{c\,z + z\,x^2 + z\,y^2 = 1, c\,y + y\,x^2 + y\,z^2 = 1, c\,x + x\,y^2 + x\,z^2 = 1\}$$

```
> _EnvExplicit := false: # stick to RootOfs
> solve( eqns, {x,y,z} );
```

$$\{z = \mathrm{RootOf}(c_Z - 1 + 2_Z^3), y = \mathrm{RootOf}(c_Z - 1 + 2_Z^3),$$
$$x = \frac{1}{c + 2\,\mathrm{RootOf}(c_Z - 1 + 2_Z^3)^2}\},$$
$$\left\{z = \%2, y = -\frac{-1 + 2\,c\,\%2 + 2\,\%2^3}{c}, x = -\frac{c^2}{-1 + 3\,c\,\%2 + 2\,\%2^3}\right\},$$
$$\left\{y = \%2, z = -\frac{-1 + 2\,c\,\%2 + 2\,\%2^3}{c}, x = -\frac{c^2}{-1 + 3\,c\,\%2 + 2\,\%2^3}\right\},$$
$$\left\{y = -\frac{1}{2}\,\frac{c^2 + c\,\%2^2 + \%2}{c\,\%2 + \%2^3 - 1}, z = \%2,\right.$$
$$\left. x = \frac{1}{2}\,\frac{c^2\,\%2^2 - 4 + 7\,c\,\%2 + 6\,\%2^3 + c^3}{\%2\,c^2 + c\,\%2^3 - 2\,c - \%2^2}\right\},$$
$$\left\{x = \frac{c + \%1^2}{\mathrm{RootOf}(_Z^2 + _Z\,\%1 + \%1^2 + c)},\right.$$
$$\left. y = \mathrm{RootOf}(_Z^2 + _Z\,\%1 + \%1^2 + c), z = \%1\right\}$$
$$\%1 := \mathrm{RootOf}(1 + c_Z + _Z^3)$$
$$\%2 := \mathrm{RootOf}(3\,c_Z^2 + c^2 + 2_Z^4 - _Z)$$

Ez a válasz csak egy strukturálatlan számhalmaz. Algebrai egyenletrendszerek megoldásánál hasznos lehet a groebner csomag **gsolve** eljárása. A **gsolve**-nak csak a nullára redukált egyenletek baloldalán álló polinomokat kell megadni.

```
> polys := map( lhs - rhs, eqns );
```

$$polys := \{c\,y + y\,x^2 + y\,z^2 - 1, c\,x + x\,y^2 + x\,z^2 - 1, c\,z + z\,x^2 + z\,y^2 - 1\}$$

```
> with( grobner ):  # load Groebner basis package
> settime := time():
> sys := gsolve( polys, {x,y,z} );
```

$$\begin{aligned}sys := [&[x - z,\, y - z,\, c\,z - 1 + 2\,z^3],\\ &[c\,x - 1 + 2\,c\,z + 2\,z^3,\, y - z,\, c^2 + 3\,c\,z^2 + 2\,z^4 - z],\\ &[x + z + y,\, y\,z + c + y^2 + z^2,\, 1 + c\,z + z^3],\\ &[x - z,\, -1 + 2\,c\,z + 2\,z^3 + c\,y,\, c^2 + 3\,c\,z^2 + 2\,z^4 - z], [\\ &2\,x - c^2 + (-c^3 - 2)\,z + 4\,c\,z^2 - 2\,z^3\,c^2,\\ &-c^2 + (-c^3 - 2)\,z + 4\,c\,z^2 - 2\,z^3\,c^2 + 2\,y,\\ &z^2\,c^2 + 1 + 2\,c\,z - 2\,z^3 + 2\,c\,z^4]]\end{aligned}$$

```
> cpu_time := (time()-settime) * seconds; # computing time
```

$$cpu_time := 117.374\,seconds$$

Olyan új polinomrendszerek listáját kaptuk, amelyek közös zérushelyei megegyeznek az eredeti rendszer megoldásaival. Ezek az új rendszerek sokszor már könyebben megoldhatók, vagy legalábbis jobban megmutatják, hogyan osztható föl az eredeti probléma kisebb részproblémákra.

```
> for s in sys do
>   solve( {op(s)}, {x,y,z} )
> od;
```

$$\begin{aligned}\{&x = \mathrm{RootOf}(c_Z - 1 + 2_Z^3),\, z = \mathrm{RootOf}(c_Z - 1 + 2_Z^3),\\ &y = \mathrm{RootOf}(c_Z - 1 + 2_Z^3)\}\end{aligned}$$

$$\{y = \%1,\, z = \%1,\, x = -\frac{-1 + 2\,c\,\%1 + 2\,\%1^3}{c}\}$$
$$\%1 := \mathrm{RootOf}(3\,c_Z^2 + c^2 + 2_Z^4 - _Z)$$

$$\begin{aligned}\{&x = -\%1 - \mathrm{RootOf}(_Z^2 + _Z\,\%1 + \%1^2 + c),\\ &y = \mathrm{RootOf}(_Z^2 + _Z\,\%1 + \%1^2 + c),\, z = \%1\}\\ &\%1 := \mathrm{RootOf}(1 + c_Z + _Z^3)\end{aligned}$$

$$\{z = \%1,\, x = \%1,\, y = -\frac{-1 + 2\,c\,\%1 + 2\,\%1^3}{c}\}$$
$$\%1 := \mathrm{RootOf}(3\,c_Z^2 + c^2 + 2_Z^4 - _Z)$$

$$\begin{aligned}\{&z = \%1,\, y = \frac{1}{2}\,c^2 + \frac{1}{2}\,\%1\,c^3 + \%1 - 2\,c\,\%1^2 + \%1^3\,c^2,\\ &x = \frac{1}{2}\,c^2 + \frac{1}{2}\,\%1\,c^3 + \%1 - 2\,c\,\%1^2 + \%1^3\,c^2\}\\ &\%1 := \mathrm{RootOf}(_Z^2\,c^2 + 1 + 2\,c_Z - 2_Z^3 + 2\,c_Z^4)\end{aligned}$$

Hasonlítsuk össze a **gsolve** eredményét az eredeti rendszer Gröbner-bázisának kiszámításával:

```
> settime := time():
> G := gbasis( polys, [x,y,z], 'plex' ):
> cpu_time := (time()-settime) * seconds;
```

$$cpu_time := 2346.394\,seconds$$

```
> factor( G[-1] );
```

$$(c\,z - 1 + 2\,z^3)\,(1 + c\,z + z^3)\,(z^2\,c^2 + 1 + 2\,c\,z - 2\,z^3 + 2\,c\,z^4)$$
$$(c^2 + 3\,c\,z^2 + 2\,z^4 - z)$$

A **gsolve** szerinti megközelítésnél valahányszor faktorizálható polinomhoz jutunk, a Maple az egyes tényezőknek megfelelő részfeladatokra bontja föl az eredeti feladatot. Ezzel növelhető az eljárás hatékonysága, ami a Gröbner-bázis módszer alkalmazásainak egyik kulcskérdése.

Harmadik példánk olyan algebrai egyenletrendszerre, melyet a Gröbner-bázis módszerrel próbálunk megoldani, a geodéziából származik. Itt csak fölvázoljuk a problémát és a Maple segítségével történő megoldását. A részletek megtalálhatók a [98]-ban.

A föld felszínén vagy annak közelében lévő P pont Descartes-féle geocentrikus x, y és z koordinátái, valamint a geocentrikus referencia-ellipszoidra vonatkozó Helmert-féle projekciójának geodétikus koordinátái, a h (magasság), a λ (földrajzi hosszúság) és a ϕ (földrajzi szélesség) közti összefüggést a következő kifejezések írják le.

(16.1) $$x = (N + h)\cos\phi\cos\lambda,$$

(16.2) $$y = (N + h)\cos\phi\sin\lambda,$$

(16.3) $$z = (N(1 - e^2) + h)\sin\phi,$$

ahol az N vertikális főgörbületi sugarat és a referencia-ellipszoid e excentricitását a következő formulák definiálják:

$$N = \frac{a}{\sqrt{1 - e^2\sin^2\phi}}$$

és

ahol a és b a referencia-ellipszoid fő- és melléktengelye. A fenti egyenletek alapján az x, y és z Descartes-féle koordinátákat közvetlenül kiszámolhatjuk a h, λ és ϕ geodétikus koordináták segítségével. Az inverz probléma sokkal nehezebb: ekkor a fönti nemlineáris egyenletrendszert kell megoldani a h, λ és a ϕ ismeretlenekre adott x, y és z mellett.

Ha megoldatjuk a Maple-lel ezt a trigonometrikus egyenletrendszert, az eredményben a földrajzi szélesség nyolcadfokú polinomok gyökeivel fölírva szerepel! De ennél okosabban is eljárhatunk. Először is a trigonometrikus mennyiségekre

bevezetett új ismeretlenek és a jólismert trigonometriai azonosságokból származó algebrai összefüggések hozzávételével rendeljünk eredeti rendszerünkhöz egy algebrai egyenletrendszert. A következő változókat fogjuk használni: $cf = \cos\phi$, $sf = \sin\phi$, $tf = \tan\phi$, $cl = \cos\lambda$ és $sl = \sin\lambda$. A négyzetgyökös kifejezésre bevezetjük az $S = \sqrt{1 - e^2 mathitsf^2}$ változót, végül legyen $d = (N + h)cf$. Így a következő egyenletrendszert kapjuk:

```
> sys := [ x - (N+h)*cf*cl, y - (N+h)*cf*sl,
> z - (N*(1-e^2)+h)*sf, cf^2 + sf^2 - 1, cl^2+sl^2 - 1,
> tf*cf - sf, N*S - a, S^2 + e^2*sf^2 - 1, (N+h)*cf - d,
> d^2 - x^2 - y^2 ];
```

$$sys := [x - (N+h)\,cf\,cl,\, y - (N+h)\,cf\,sl,\, z - (N\,(1-e^2)+h)\,sf,\\ cf^2 + sf^2 - 1,\, cl^2 + sl^2 - 1,\, tf\,cf - sf,\, N\,S - a,\, S^2 + e^2\,sf^2 - 1,\\ (N+h)\,cf - d,\, d^2 - x^2 - y^2]$$

Kiszámítjuk a változók

$$N > S > x > y > h > cl > sl > cf > sf > tf$$

valódi lexikografikus rendezésének megfelelő Gröbner-bázist:

```
> with( grobner ): # load Groebner basis package
> vars := [N,S,x,y,h,cl,sl,cf,sf,tf]:
> gsys := gbasis( sys, vars, 'plex' ):
```

A teljes Gröbner-bázis túl nagy ahhoz, hogy kiírassuk. Egyébként is csak a *tf* változót tartalmazó egyváltozós polinom érdekel bennünket, ami várhatóan a Gröbner-bázis utolsó polinomja lesz:

```
> collect( gsys[-1], tf ); # get the polynomial in tf
```

$$-z^2 + (2\,z\,d - 2\,d\,z\,e^2)\,tf^3 + 2\,z\,d\,tf + (-d^2 - z^2 + e^2\,z^2 + a^2\,e^4)\,tf^2\\ + (-d^2 + d^2\,e^2)\,tf^4$$

```
> map( convert, - ", sqrfree ); # rewrite the polynomial
```

$$z^2 + 2\,(-1 + e^2)\,z\,d\,tf^3 - 2\,z\,d\,tf - (-d^2 - z^2 + e^2\,z^2 + a^2\,e^4)\,tf^2\\ - (-1 + e^2)\,d^2\,tf^4$$

```
> sort( subs( e^2*z^2 = (e^2-1)*z^2 + z^2, " ), tf );
```

$$-(-1 + e^2)\,d^2\,tf^4 + 2\,(-1 + e^2)\,z\,d\,tf^3 - (-d^2 + (-1 + e^2)\,z^2 + a^2\,e^4)\,tf^2\\ - 2\,z\,d\,tf + z^2$$

Végül tehát a $\tan\phi$ olyan negyedfokú polinomját kaptuk, amely már analitikusan megoldható (v. ö. [150]).

Ugyanerre az eredményre jutottunk volna a **finduni** eljárással is:

```
> finduni(tf,sys,vars);
```

$$-z^2 + (2\,z\,d - 2\,d\,z\,e^2)\,tf^3 + 2\,z\,d\,tf + (-d^2 - z^2 + e^2\,z^2 + a^2\,e^4)\,tf^2\\ + (-d^2 + d^2\,e^2)\,tf^4$$

Az eljárás a változók teljes fokszáma szerinti rendezést alkalmazza a Gröbner-bázis kiszámításához, majd ezt fölhasználva konstruálja meg a polinomok által generált ideálnak a (*tf*-fel) fölírt legkisebb fokszámú egyváltozós polinomját. Ebben a speciális esetben a számítási idő két és félszerese az előzőének, amikor lexikografikus rendezéssel dolgoztunk.

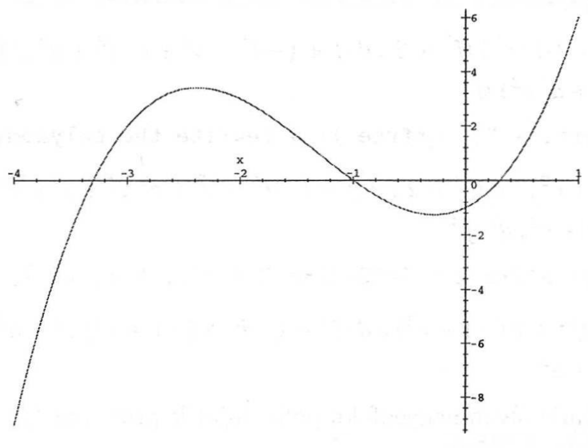
Az eddig látott példák alapján már bizonyára kialakult valamilyen elképzelésünk a Gröbner-bázis módszer erősségéről; akár úgy is, mint az eliminációs vagy a Ritt–Wu féle karakterisztikus halmazok módszerének lehetséges alternatívájáról. (Az utóbbi Dongming Wang által implementált és a [186]-ban publikált változata megtalálható a Maple osztott könyvtárában.) Mégegyszer emlékeztetünk arra, hogy a Gröbner-bázis algoritmus alkalmazásánál komoly hátrány lehet a jelentős idő és memóriakomplexitás.

16.6. Egyenlőtlenségek

Az alábbi példák alapján az Olvasó képet alkothat a Maple egyenlőtlenségek és egyenlőtlenség-rendszerek megoldásával kapcsolatos képességeiről. A sikeres megoldás szükséges föltétele, hogy egyértelműen elrendezhetők legyenek azok a pontok, ahol az egyenlőtlenségek egyenlőségbe mennek át.

Tekintsük az $x \longmapsto x^3 + 4x^2 + 2x - 1$ polinomot, amelynek grafikonja a 16.4. ábrán látható.

```
> plot( x^3 + 4*x^2 + 2*x - 1, x = -4..1 );
```



16.4. ábra: Az $x^3 + 4x^2 + 2x - 1$ grafikonja a $(-4, 1)$-en

Határozzuk meg, hol pozitív a függvény.

```
> solve( x^3 + 4*x^2 + 2*x - 1 > 0, x );
```

$$\text{RealRange}(\text{Open}(-\frac{3}{2} - \frac{1}{2}\sqrt{13}), \text{Open}(-1)),$$
$$\text{RealRange}(\text{Open}(-\frac{3}{2} + \frac{1}{2}\sqrt{13}), \infty)$$

Az egyismeretlenes egyenletekhez hasonlóan a megoldás nem tartalmaz változókat, de most a `RealRange` jelöléssel van fölírva. Ha a feladatot egyenlőtlenségek halmazaként adjuk meg, (az egyenletekhez hasonlóan) az ismeretlent tartalmazó egyenlőtlenségek formájában kapjuk a megoldást.

```
> solve( { x^3 + 4*x^2 + 2*x - 1 > 0 }, {x} );
```

$$\{-\frac{3}{2} - \frac{1}{2}\sqrt{13} < x, x < -1\}, \{-\frac{3}{2} + \frac{1}{2}\sqrt{13} < x\}$$

Az egyenlőtlenség körüli halmaz-zárójelek elhagyhatók, ekkor is megoldáshalmazokat kapunk:

```
> solve( abs(x)*x + exp(x)>=0, {x} );
```

$$\{-2\,\text{LambertW}(\frac{1}{2}) \le x\}$$

```
> solve( a*x + b >= c, {x} );
```

$$\{-\text{signum}(a)\,x \le -\frac{\text{signum}(a)\,(c-b)}{a}\}$$

A **solve** egyenlőtlenségrendszerekre is alkalmazható:

```
> solve( { x^2 - x = 0, x<>0 }, x ); # nonzero solution
```

$$\{x = 1\}$$

```
> _EnvExplicit := true:
> solve( {x^4-9, x^2-3<>0} );
```

$$\{x = I\sqrt{3}\}, \{x = -I\sqrt{3}\}$$

```
> solve( { x<1/3, y<1/2, x^2+y^2=1 }, {x,y} );
```

$$\{y < -\frac{2}{3}\sqrt{2}, x = \sqrt{1-y^2}\}, \{y < -\frac{2}{3}\sqrt{2}, x = -\sqrt{1-y^2}\},$$
$$\{-\frac{2}{3}\sqrt{2} < y, x = \sqrt{1-y^2}, y < \frac{1}{2}\}, \{-\frac{2}{3}\sqrt{2} < y, x = -\sqrt{1-y^2}, y < \frac{1}{2}\}$$

```
> evalf(["]);
```

$$[\{x = \sqrt{1. - 1.\,y^2}, y < -.9428090414\},$$
$$\{x = -1.\,\sqrt{1. - 1.\,y^2}, y < -.9428090414\},$$
$$\{x = \sqrt{1. - 1.\,y^2}, -.9428090414 < y, y < .5000000000\},$$
$$\{x = -1.\,\sqrt{1. - 1.\,y^2}, -.9428090414 < y, y < .5000000000\}]$$

16.7. Numerikus megoldási módszerek

Egyenletek vagy egyenletrendszerek megoldásainak numerikus közelítésére az **fsolve** Maple eljárást alkalmazhatjuk. Használata hasonló a **solve**-hoz.

```
> x^7 - 2*x^6 - 4*x^5 - x^3 + x^2 + 6*x + 4;
```

$$x^7 - 2x^6 - 4x^5 - x^3 + x^2 + 6x + 4$$

```
> fsolve(");
```

$$-1.236067977, 1.167303978, 3.236067977$$

A 16.1. táblázatban fölsorolt kiegészítő opciókat is megadhatjuk:

Opció	Jelentése
`complex`	komplex értékű gyök(ök)
`a..b`	a valós számok ezen tartományában keresünk gyököt
`maxsols=`*n*	a megoldások maximális száma
`fulldigits`	`Digits` pontosságú lebegőpontos számítások

16.1. táblázat: Az **fsolve** opciói

Néhány példa:

```
> fsolve( "" , x , complex );
```

$$-1.236067977, -.7648844336 - .3524715460\,I,$$
$$-.7648844336 + .3524715460\,I, .1812324445 - 1.083954101\,I,$$
$$.1812324445 + 1.083954101\,I, 1.167303978, 3.236067977$$

```
> fsolve( """ , x , 0..2 );
```

$$1.167303978$$

Az utóbbi esetben arra kértük a Maple-t, hogy csak 0 és 2 közti valós megoldásokat próbáljon keresni.

Algebrai egyenleteknél az **fsolve** általában (de nem mindig) az összes valós megoldást kiszámítja, sőt a `komplex` opció használata esetén az összes komplexet is. Más típusú egyenletek esetében rendszerint megelégszik egyetlen megoldás meghatározásával:

```
> eqn := sin(x) = x/2;
```

$$eqn := \sin(x) = \frac{1}{2}\,x$$

```
> fsolve( eqn, x );
```

$$1.895494267$$

```
> fsolve( eqn, x, 0.1 ..  infinity );
```

$$1.895494267$$

```
> fsolve( eqn, x, -0.1 ..  0.1 );
```

$$0$$

```
> fsolve( eqn, x, -infinity .. -0.1 );
```

$$-1.895494267$$

Az **fsolve** működésének alapja két algoritmus: a (többdimenziós) Newton-módszer, illetve ha ez nem vezet eredményre, akkor a (többdimenziós) szelő-módszer (v. ö. [174]).

A **realroot** eljárás a Descartes-féle jelszabályt alkalmazza egyváltozós polinomok valós gyökeit szétválasztó intervallumok listájának előállítására. Kiegészítő opcióként megadható az intervallumok hossza is.

```
> readlib( realroot ):
> x^7 - 2*x^6 - 4*x^5 - x^3 + x^2 + 6*x + 4;
```

$$x^7 - 2\,x^6 - 4\,x^5 - x^3 + x^2 + 6\,x + 4$$

```
> realroot(");
```

$$[[0, 2], [2, 4], [-2, -1]]$$

```
> realroot( "", 1/100 );
```

$$\left[\left[\frac{149}{128}, \frac{75}{64}\right], \left[\frac{207}{64}, \frac{415}{128}\right], \left[\frac{-159}{128}, \frac{-79}{64}\right]\right]$$

A **sturm** eljárással valamely polinom adott intervallumba eső valós gyökeinek számát határozhatjuk meg. Elméleti alapja a Sturm-tétel (v. ö. [138]).

```
> readlib(sturm):
> x^7 - 2*x^6 - 4*x^5 - x^3 + x^2 + 6*x + 4;
```

$$x^7 - 2\,x^6 - 4\,x^5 - x^3 + x^2 + 6\,x + 4$$

```
> sturm( ", x, -infinity,infinity ); # three real roots
```

$$3$$

```
> sturm( "", x, 0, infinity ); # two positive roots
```

$$2$$

```
> sturm( """, x, 2, 4 ); # one root between 2 and 4
```

$$1$$

16.8. A Maple további egyenletmegoldó rutinjai

isolve

Az **isolve** eljárással egyenletek vagy egyenletrendszerek egész megoldásait kereshetjük meg. A következő példa ennek alkalmazása a dimenzióanalízisre.

A levegőben gyorsan mozgó objektum F légellenállása függ az objektum V sebességétől, d átmérőjétől, a levegő ρ sűrűségétől, a c hangsebességtől, és a ν kinematikai viszkozitástól. Ezen mennyiségek dimenziói kifejezhetők az m tömeg, az l hosszúság és a t idő dimenzióiból. Az F, V, d, ρ, c és a ν dimenziója a következő módon határozható meg:

```
> nondimensional := force^f * speed^v * diameter^d
>    * density^r * acoustic_velocity^c
```

```
>     * kinematic_viscosity^m;
```

$$nondimensional := force^f\, speed^v\, diameter^d\, density^r\, acoustic_velocity^c\, kinematic_viscosity^m$$

```
> subs( { force = M*L/T^2, speed = L/T, diameter = L,
>   density = M/L^3, acoustic_velocity = L/T,
>   kinematic_viscosity = L^2/T }, nondimensional );
```

$$\left(\frac{M\,L}{T^2}\right)^f \left(\frac{L}{T}\right)^v L^d \left(\frac{M}{L^3}\right)^r \left(\frac{L}{T}\right)^c \left(\frac{L^2}{T}\right)^m$$

```
> simplify( ", 'symbolic' );
```

$$M^{(f+r)}\, L^{(f+v+d-3\,r+c+2\,m)}\, T^{(-2\,f-v-c-m)}$$

```
> eqns := { seq( op(2,i)=0, i= " ) };
```

$$eqns := \{f+r=0, f+v+d-3r+c+2m=0, -2f-v-c-m=0\}$$

```
> isolve("); # find all integral solutions
```

$$\{f=-_N3, v=-_N1-_N2+2_N3, d=-_N1+2_N3, m=_N1, c=_N2, r=_N3\}$$

```
> subs( ", nondimensional );
```

$$force^{(-_N3)}\, speed^{(-_N1-_N2+2_N3)}\, diameter^{(-_N1+2_N3)}\, density^{_N3}\, acoustic_velocity^{_N2}\, kinematic_viscosity^{_N1}$$

Néhány trükk alkalmazásával csoportosíthatjuk az azonos kitevőjű hatványokat:

```
> expand( ln(") );
```

$$-_N3\ln(force) - \ln(speed)\,_N1 - \ln(speed)\,_N2 + 2\ln(speed)\,_N3 - \ln(diameter)\,_N1 + 2\ln(diameter)\,_N3 + _N3\ln(density) + _N2\ln(acoustic_velocity) + _N1\ln(kinematic_viscosity)$$

```
> [ coeffs( ", {_N1,_N2,_N3}, 'vars' ) ];
```

$$[-\ln(speed) + \ln(acoustic_velocity), -\ln(force) + 2\ln(diameter) + \ln(density) + 2\ln(speed), -\ln(diameter) - \ln(speed) + \ln(kinematic_viscosity)]$$

```
> combine( ", ln, anything, 'symbolic' );
```

$$\left[\ln\left(\frac{acoustic_velocity}{speed}\right), \ln\left(\frac{diameter^2\, density\, speed^2}{force}\right), \ln\left(\frac{kinematic_viscosity}{diameter\, speed}\right)\right]$$

```
> zip( (a,b) -> exp(a)^b, ", [vars] );
```

$$\left[\left(\frac{acoustic_velocity}{speed}\right)^{_N2}, \left(\frac{diameter^2\, density\, speed^2}{force}\right)^{_N3}, \left(\frac{kinematic_viscosity}{diameter\, speed}\right)^{_N1}\right]$$

```
> convert( ", `*` );
```

$$\left(\frac{acoustic_velocity}{speed}\right)^{_N2} \left(\frac{diameter^2\, density\, speed^2}{force}\right)^{_N3} \left(\frac{kinematic_viscosity}{diameter\, speed}\right)^{_N1}$$

Három egész paraméterrel meghatározott dimenzió nélküli változók családját kaptuk. Ezek kétféle kombinációja jólismert az aerodinamikából és a folyadékok dinamikájából.

```
> subs( _N1=0, _N2=-1, _N3=0, " ); # the Mach number
```

$$\frac{speed}{acoustic_velocity}$$

```
> subs( _N1=-1, _N2=1, _N3=0, "" ); # the Reynold's number
```

$$\frac{acoustic_velocity\, diameter}{kinematic_viscosity}$$

msolve

Használhatunk moduláris aritmetikát is. Következő példánk a Pell-egyenlet harmadfokú analógja $\mathbb{Z}_7$ fölött.

```
> msolve( y^2 = x^3 - 28, 7 );
```

$$\{y=0, x=0\}, \{y=6, x=4\}, \{y=6, x=1\}, \{y=6, x=2\}, \{y=1, x=4\}, \{y=1, x=1\}, \{y=1, x=2\}$$

rsolve

A Maple meg tud oldani rekurrens egyenleteket is. Ehhez olyan standard technikákat használ, mint a generátorfüggvények, a z-transzformáció, valamint a helyettesítéseken és a karakterisztikus egyenleteken alapuló módszerek. Néhány példa:

- Általánosított Fibonacci-polinomok

```
> rsolve( { f(n+2) = f(n+1) + f(n),
> f(0)=0, f(1)=1 }, f(n) ); # Fibonacci numbers
```

$$\frac{\left(-1+\frac{1}{5}\sqrt{5}\right)\left(-\frac{2}{-\sqrt{5}+1}\right)^{n}}{-\sqrt{5}+1}+\frac{\left(-1-\frac{1}{5}\sqrt{5}\right)\left(-\frac{2}{\sqrt{5}+1}\right)^{n}}{\sqrt{5}+1}$$

```
> rsolve( { f(n+2) = x*f(n+1) + y*f(n),
> f(0)=0, f(1)=1 }, f(n) );
```

$$-\frac{\left(x^2+4y-x\sqrt{x^2+4y}\right)\left(-2\frac{y}{x-\sqrt{x^2+4y}}\right)^{n}}{\left(x^2+4y\right)\left(x-\sqrt{x^2+4y}\right)}$$

$$-\frac{\left(x\sqrt{x^2+4y}+x^2+4y\right)\left(-2\frac{y}{x+\sqrt{x^2+4y}}\right)^{n}}{\left(x^2+4y\right)\left(x+\sqrt{x^2+4y}\right)}$$

```
> normal( ", 'expanded' );
```

$$\frac{(-2)^n y^n\left(\frac{1}{x-\sqrt{x^2+4y}}\right)^{n}-(-2)^n y^n\left(\frac{1}{x+\sqrt{x^2+4y}}\right)^{n}}{\sqrt{x^2+4y}}$$

```
> # 5th generalized Fibonacci polinomial
> normal( subs( n=5," ), 'expanded' );
```

$$x^4+3\,y\,x^2+y^2$$

Ennek kapcsán megjegyezzük, hogy $n < 100$-ra igaz az a sejtés, amely szerint $f(n)$ akkor és csak akkor irreducibilis, ha n prím.

- A Gauss-elimináció komplexitása

```
> rsolve( { T(n) = T(n-1) + n^2, T(1)=0 }, T(n) );
```

$$2\,(n+1)\left(\frac{1}{2}n+1\right)\left(\frac{1}{3}n+1\right)-3\,(n+1)\left(\frac{1}{2}n+1\right)+n$$

```
> factor(");
```

$$\frac{1}{6}(n-1)\left(2n^2+5n+6\right)$$

- Az összefésüléses rendezés („merge sort") komplexitása

```
> rsolve( { T(n) = 2*T(n/2) + n-1, T(1)=0 }, T(n) );
```

$$n\left(2\left(\frac{1}{2}\right)^{\left(\frac{\ln(n)}{\ln(2)}+1\right)}+\frac{\ln(n)}{\ln(2)}-1\right)$$

```
> simplify(");
```

$$-\frac{-\ln(2)-\ln(n)\,n+n\ln(2)}{\ln(2)}$$

- A Karacuba-féle szorzási algoritmus komplexitása

```
> rsolve( { T(n) = 3*T(n/2) + n, T(1)=1}, T(n) );
```

$$n^{\left(\frac{\ln(3)}{\ln(2)}\right)} + n^{\left(\frac{\ln(3)}{\ln(2)}\right)}\left(-3\left(\frac{2}{3}\right)^{\left(\frac{\ln(n)}{\ln(2)}+1\right)} + 2\right)$$

```
> simplify(");
```

$$3\,n^{\left(\frac{\ln(3)}{\ln(2)}\right)} - 2\,n$$

Az összegzési problémákat néha csak részlegesen tudja megoldani a Maple.

```
> rsolve( a(n+1) = ln(n+1)*a(n) + 1, a(n) );
```

$$\left\{ \mathrm{a}(n) = \left(\prod_{_n1=1}^{n-1} \ln(_n1+1)\right)\left(\sum_{_n2=1}^{n-1}\left(1\Big/\prod_{_n1=1}^{_n2}\ln(_n1+1)\right) + \ln(2)\,\mathrm{a}(0) + 1\right), \mathrm{a}(0) = \mathrm{a}(0) \right\}$$

A megoldás aszimptotikus viselkedéséről néha még akkor is tud valami információt nyújtani a rendszer, ha az **rsolve** nem talált zárt alakban fölírható megoldást.

```
> rsolve( u(n+1) = ln( u(n)+1 ), u(n) );
```

$$\mathrm{rsolve}(\mathrm{u}(n+1) = \ln(\mathrm{u}(n)+1), \mathrm{u}(n))$$

```
> asympt( ", n, 4 );
```

$$\frac{2}{n} + \frac{_C + \frac{2}{3}\ln(n)}{n^2} + \mathrm{O}\left(\frac{1}{n^3}\right)$$

Befejező példánk a Knuth által [118]-ban kitűzött probléma megoldását tartalmazza. Fejezzük ki az

$$x_0 = a, \quad x_1 = b, \quad x_{n+2} = x_{n+1} + x_n/(n+1), \qquad n = 0, 1, 2, \ldots$$

rekurrens egyenlet megoldását n ismert függvényeivel, és írjuk föl a megoldás aszimptotikus alakját is.

```
> infolevel[rsolve]:= 5:
> rsolve( { x(n+2) = x(n+1) + x(n)/(n+1),
> x(0)=a, x(1)=b }, x(n) );
```

$$_C_1\,(n+1)$$

Túl optimisták voltunk. A Maple csak a megoldás aszimptotikus alakját tudta meghatározni. Az `LREtools` csomag új rutinjai, melyek lineáris rekurziókat kezelnek, ebben az esetben nem vezettek eredményre. Próbáljunk segíteni a Maple-nek. Először számítsuk ki a rekurrens egyenlet z-transzformáltját (kicsit másként fölírt, de az eredetivel ekvivalens alakban).

```
> readlib( ztrans ): # load library function
> x(0) := a: x(1) := b:
> subs( ztrans(x(n),n,z) = F(z),
```

```
> ztrans( (n+1)*x(n+2)=(n+1)*x(n+1)+x(n), n, z ) );
```

$$-z^2\,\mathrm{F}(z) - z^3\,(\frac{\partial}{\partial z}\,\mathrm{F}(z)) + a\,z^2 = -z^2\,(\frac{\partial}{\partial z}\,\mathrm{F}(z)) + \mathrm{F}(z)$$

Ezt a differenciálegyenletet kell megoldanunk.

```
> dsolve( ", F(z) );
```

$$\mathrm{F}(z) = \frac{z\,(e^{(\frac{1}{z})}\,z\,a + _C1)\,e^{(-\frac{1}{z})}}{1 - 2\,z + z^2}$$

A számítás utolsó lépése, az inverz z-transzformált meghatározása a legnehezebb.

```
> expand(");
```

$$\mathrm{F}(z) = \frac{z^2\,a}{1 - 2\,z + z^2} + \frac{z\,_C1}{e^{(\frac{1}{z})}\,(1 - 2\,z + z^2)}$$

```
> subs( invztrans(F(z),z,n)=x(n), invztrans(",z,n) );
```

$$\mathrm{x}(n) = a\,n + a + _C1\ \mathrm{invztrans}\left(\frac{z}{e^{(\frac{1}{z})}\,(1 - 2\,z + z^2)},\, z,\, n\right)$$

```
> simplify( factor(") );
```

$$\mathrm{x}(n) = a\,n + a + _C1\ \mathrm{invztrans}\left(\frac{z\,e^{(-\frac{1}{z})}}{(-1 + z)^2},\, z,\, n\right)$$

```
> f := subs( body = invztrans(exp(-1/z),z,k), k->body );
```

$$f := k \to \frac{(-1)^k}{k!}$$

```
> g := subs( body = invztrans(z/(z-1)^2,z,k), k->body );
```

$$g := k \to k$$

A kívánt z-transzformáltat innen konvolúcióval nyerjük.

```
> lhs(""") = subsop( 3 = _C1*Sum(f(n-k)*g(k),k=0..n),
> rhs("""));
```

$$\mathrm{x}(n) = a\,n + a + _C1\left(\sum_{k=0}^{n} \frac{(-1)^{(n-k)}\,k}{(n-k)!}\right)$$

```
> (value@subs)( n=1, " );
```

$$b = 2\,a + _C1$$

```
> solve(",_C1);
```

$$b - 2\,a$$

```
> subs( _C1=", """ );
```

$$\mathrm{x}(n) = a\,n + a + (b - 2\,a)\left(\sum_{k=0}^{n} \frac{(-1)^{(n-k)}\,k}{(n-k)!}\right)$$

A Maple még az összeg zárt alakját is meg tudja mondani.

```
> value(");
```

$$x(n) = a\,n + a + \frac{(b-2\,a)\,(-1)^{(n-1)}\,\mathrm{hypergeom}(\,[-n+1,\,2\,],\,[\,],\,1)}{(n-1)!}$$

A `genfunc` csomag függvényei segítségével racionális generátorfüggvényekkel dolgozhatunk. Hasznos segédeszköz lehet, ha a generátorfüggvények módszerét „manuálisan" próbáljuk rekurrens egyenletekre alkalmazni.

match

Jelenleg a Maple csak korlátozott mintaillesztési lehetőségeket biztosít. A **match** függvény egyváltozós kifejezésnek megadott mintával való egyezését vizsgálja. Ilyen egyszerű feladat például a következő: megadható-e olyan a és b, hogy $x^2 + 2 * x + 3 = (x + a)^2 + b$ legyen minden x-re.

```
> match( x^2 + 2*x + 3 = (x+a)^2 + b, x, 'sol' );
```

$$true$$

A Maple megerősíti, hogy elérhető egyezés. A megfelelő a és b értékeket a harmadik paraméterként megadott `sol` változóban kapjuk.

```
> sol;
```

$$\{a = 1,\, b = 2\}$$

Lényeges, hogy pontosan megértsük az algebrai mintaillesztést végző **match** és a **typematch** közti különbséget. Az utóbbi csupán az objektumok alakját vizsgálja.

```
> typematch( (x+1)^2 + 2,
>   (a::anything) &+ (b::anything) );
```

$$true$$

```
> a,b;
```

$$(x+1)^2,\, 2$$

```
> typematch( (x+1)^2 + 2,
>   (((u::anything) &+ (v::anything))^2) &+ (w::anything) );
```

$$true$$

```
> typematch( x^2 + 2*x + 3,
>   (((p::anything) &+ (q::anything))^2) &+ (r::anything) );
```

$$false$$

16.9. Gyakorlatok

1. Számítsuk ki az a hatodfokú polinomot, amely átmegy a $(-5,-120)$, $(-3,48)$, $(-2,36)$, $(1,120)$, $(4,2400)$, $(10,220380)$ és a $(12,57408)$ pontokon.

2. Vizsgáljuk meg, hogy a

$$3x^2+3y^2+6xy+6x+6y+2$$

polinom fölírható-e

$$a(x+by+c)^n+d$$

alakban alkalmasan választott a, b, c, d és n értékekkel.

3. Tekintsük a következő két kémiai reakciót.

(a) A p, q, r, s, t, u és v mely értékeire lesz a

$$p\,\mathrm{KMnO_4}+q\,\mathrm{H_2SO_4}+r\,\mathrm{H_2C_2O_4}\longrightarrow$$
$$s\,\mathrm{K_2SO_4}+t\,\mathrm{MnSO_4}+u\,\mathrm{H_2O}+v\,\mathrm{CO_2}$$

reakcióegyenlet kiegyensúlyozott?

(b) A p, q, r, s és t mely értékére lesz a következő reakcióegyenlet

$$p\,\mathrm{CO}+q\,\mathrm{CO_2}+r\,\mathrm{H_2}\longrightarrow s\,\mathrm{cH_4}+t\,\mathrm{H_2O}$$

kiegyensúlyozott?

4. Oldjuk meg a következő egyenletet a **solve** és az **fsolve** segítségével:

$$48x^5+8x^4-6x^3+114x^2-37x+18=0$$

5. Oldjuk meg a következő egyenleteket.

(a) $(x+1)^{(x+a)}=(x+1)^2$

(b) $x+(1+x)^{1/2}+(2+x)^{1/3}+(3+x)^{1/4}=5$

(c) $2\arctan x=\arctan\left(\dfrac{2x}{1-x^2}\right)$

(d) $2-\sin(1-x)=2x$

6. Az (x,y,z) Descartes-féle koordináták könnyen kifejezhetők az (r,θ,ϕ) szférikus koordinátákkal:

$$\begin{aligned} x &= r\cos\theta\sin\phi \\ y &= r\sin\theta\sin\phi \\ z &= r\cos\phi \end{aligned}$$

Fejezzük ki a szférikus koordinátákat a Descartes-félékkel.

7. Oldjuk meg az $\{x^2 + y^2 = 5, xy = y^2 - 2\}$ egyenletrendszert a **solve** és az **fsolve** segítségével.

8. Oldjuk meg az x, y és z valós ismeretlenekre a következő algebrai egyenletrendszert (a valós konstanst jelöl):

$$\{\begin{aligned} z^2 - x^2 - y^2 + 2ax + 2az - a^2 &= 0, \\ yz - ay - ax + a^2 &= 0 \\ -2a + x + y &= 0\,\} \end{aligned}$$

9. Legyen f az $x_0^3 + x_1^3 + x_2^3 + x_3^3$ homogén polinom. Ez a $\mathbb{P}^3$ projektív térben az

$$\{x_0 : x_1 : x_2 : x_3 \in \mathbb{P}^3 \mid f(x_0, x_1, x_2, x_3) = 0\}$$

Fermat-felületet definiálja.

(a) A felületen 27 egyenes van. Határozzuk meg ezeket az egyeneseket a Maple fölhasználásával. Útmutatás: az $x_0 : x_1 : x_2 : x_3$ és az $y_0 : y_1 : y_2 : y_3$ pontokon átmenő egyenest az ún. Plücker koordinátákkal így írhatjuk föl:

$$p^{ij} := x_i y_j - x_j y_i, \qquad \text{ahol} \quad i, j = 0, 1, 2, 3 \quad \text{és} \quad i \neq j.$$

Az ebben a koordinátarendszerben megadott egyenes akkor és csak akkor van rajta a felületen, ha

$$\begin{aligned} &f(p^{01}u_1 + p^{02}u_2 + p^{03}u_3,\, p^{10}u_0 + p^{12}u_2 + p^{13}u_3, \\ &p^{20}u_0 + p^{21}u_1 + p^{23}u_3,\, p^{30}u_0 + p^{31}u_1 + p^{32}u_2) = 0 \end{aligned}$$

minden $u_0, u_1, u_2, u_3 \in \mathbb{R}$-re, továbbá $p^{01}p^{23} + p^{02}p^{31} + p^{03}p^{12} = 0$.

(b) Határozzuk meg a Fermat-felület szinguláris pontjait. Emlékeztetésül egy pont akkor szinguláris, ha ott a függvény és az össze parciális derivált értéke 0.

10. Oldjuk meg a következő egyenlőtlenségeket.

(a) $|x - 3| \cdot |3 - x| > |x|$

(b) $|x^3 - x^2 - x - 1| > \dfrac{1}{|x^2 - 1|}$

11. Oldjuk meg az $a_{n+1} = (8/5)\, a_n - a_{n-1}$, $a_0 = 0$, $a_1 = 1$ rekurrens egyenletet.

12. Oldjuk meg az $a_{n+1} = 3n\, a_n - 2n(n-1)a_{n-1}$, $a_1 = 5$, $a_2 = 54$ rekurrens egyenletet.

17. Differenciálegyenletek

A Maple sok közönséges differenciálegyenlet analitikus megoldását elő tudja állítani explicit vagy implicit formában. A közönséges differenciálegyenleteket megoldó **dsolve** eljárás egy sor hagyományos technikán alapul: Laplace-transzformáció, integráló tényezők keresése stb. A **pdesolve** eljárás parciális differenciálegyenleteket old meg a karakterisztikák módszerével és más klasszikus módszerekkel. A `liesym` csomagban a parciális differenciálegyenleteknél szokásos Lie szimmetria módszereket is implementálták. Ezekkel könnyebben generálhatunk további megoldásokat a parciális differenciálegyenlet valamely partikuláris megoldásából. Közönséges differenciálegyenletekre elérhető a Taylor-sor módszer, a hatványsorok módszere és más közelítő módszerek is. Ha mindezek nem vezetnek eredményre, még mindig próbálkozhatunk a Runge–Kutta és más numerikus módszerek használatán alapuló numerikus megoldó rutinokkal. A `DEtools` csomag a megoldások ábrázolását és a (független vagy akár a függő) változók transzformációit támogató eljárásokat tartalmaz. Ezen túl a perturbációs (például a Poincaré–Lindstedt vagy a többszörös skálázásos) módszer végrehajtásához szükséges segédeszközöket is megkapjuk a Maple-től. Ez a fejezet a Maple differenciálegyenletek tanulmányozására szolgáló eszközeit tárgyalja. Sok példánk az alkalmazott matematika területéről származik.

17.1. Vessünk egy pillantást a differenciálegyenletekre

Idézzük föl a *közönséges differenciálegyenlet* (KDE) fogalmát. Ez adott intervallumon teljesülő olyan

$$F(y, y', y'', \ldots, y^{(n)}, x) = 0$$

alakú egyenletet jelent, amelyben F az $\mathbb{R}^{n+2}$ valamely részhalmazán értelmezett valós függvény, $y', y'', \ldots, y^{(n)}$ pedig az $y = y(x)$ ismeretlen függvény deriváltjait jelöli. A KDE *n-ed rendű*, ha az F függvény nem függ $(n+1)$-dik argumentumától. Ha F lineáris első $n+1$ argumentumában,

$$a_n(x)y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y + a(x) = 0$$

alakban is fölírható, ekkor a KDE *lineáris*. Ha F polinomfüggvény, az n-ed rendű KDE *fokán* F $(n+1)$-dik argumentumának kitevőjét értjük. Rövidebben szólva a KDE rendje k, ha a benne előforduló deriváltak közül a k-dik a legmagasabb rendű, továbbá a KDE foka ennek a deriváltnak a kitevője. Néhány példa:

$y'' - x^2y - x^3 = 0$		másodrendű elsőfokú lineáris KDE,
$y'' - y^3 = 0$		másodrendű elsőfokú nemlineáris KDE,
$(y'')^3 - y = 0$		másodrendű harmadfokú nemlineáris KDE.

A matematikusok differenciálegyenleteket megoldó módszerek széles választékát dolgozták ki. Az érdeklődő Olvasónak a [135]-ben található áttekintést ajánljuk a számítógépes algebra alkalmazásáról a közönséges differenciálegyenletek megoldásában. A módszerek egy részét a Maple-ben is implementálták. Ismét megemlítjük, hogy az összes módszert a **dsolve** eljárás tartalmazza. A Maple-re bízhatjuk a differenciálegyenletet megoldó módszer kiválasztását, de mi magunk is választhatunk. Ha például a Laplace-transzformáltak módszerét kívánjuk alkalmazni, vagy a megoldás Taylor-sorát szeretnénk megtalálni, csak ezt kell megmondanunk a rendszernek.

Őszintén szólva bármilyen imponálóak a Maple közönséges differenciálegyenletek megoldásával kapcsolatos lehetőségei, igencsak szegényesnek tűnnek a téma óriási szakirodalmához képest. Legtöbbször csak az elsőfokú és a háromnál alacsonyabb rendű KDE-ket tudjuk megoldatni a rendszerrel. A **dsolve** ismer bizonyos speciális függvényeket is. Mindazonáltal a Maple értékes segítséget nyújthat a differenciálegyenletek megoldásánál. A következő részekben az egyes lehetőségeket az alkalmazott matematikából vett számos példa kapcsán taglaljuk.

17.2. Analitikus megoldások

Tekintsük az elsőrendű, elsőfokú

$$xy' = y\ln(xy) - y$$

differenciálegyenletet. Ha nem ragaszkodunk a KDE valamelyik kitüntetett megoldási módszeréhez, elegendő simán meghívni a **dsolve**-ot:

```
> ODE := x*diff(y(x),x) = y(x)*ln(x*y(x)) - y(x);
```

$$ODE := x\left(\frac{\partial}{\partial x}\,\mathrm{y}(x)\right) = \mathrm{y}(x)\ln(x\,\mathrm{y}(x)) - \mathrm{y}(x)$$

```
> dsolve( ODE, y(x) );
```

$$x = _C1 \ln(x) + _C1 \ln(y(x))$$

Kényelmesebben kezelhető és természetesebb jelölést vezethetünk be a Maple **alias** konstrukciójával. Alkalmazzuk ezt ugyanarra a feladatra.

```
> alias( y=y(x) ):
> ODE := x*diff(y,x) = y*ln(x*y) - y;
```

$$ODE := x\,(\frac{\partial}{\partial x}\,y) = y\ln(x\,y) - y$$

```
> dsolve( ODE, y );
```

$$x = _C1 \ln(x) + _C1 \ln(y)$$

Látható, hogy a Maple implicit megoldás keresése mellett döntött. Ha az y függvényre explicit megoldást szeretnénk kapni, használhatjuk a **solve**-ot, de segítségünkre lehet az **isolate** is.

```
> readlib(isolate)(",y);
```

$$y = e^{\left(\frac{x - _C1 \ln(x)}{_C1}\right)}$$

Mindezt azonban már az elején is közölhettük volna a Maple-lel.

```
> dsolve( ODE, y, explicit = true );
```

$$y = e^{\left(\frac{x - _C1 \ln(x)}{_C1}\right)}$$

Az általános megoldás ennél egyszerűbben is fölírható.

```
> expand(");
```

$$y = \frac{e^{\left(\frac{x}{_C1}\right)}}{x}$$

```
> subs( _C1=1/c, " );
```

$$y = \frac{e^{(c\,x)}}{x}$$

A KDE fönti megoldása már könnyen ellenőrizhető (és ezt ajánlatos is elvégezni, ahányszor csak módunkban áll).

```
> testeq( subs( ", ODE ) );
```

$$\mathit{true}$$

Második példaként tekintsük az

$$y\,y' = \sqrt{y^2 + 2y + 1}$$

differenciálegyenletet. Megnöveljük az `infolevel[dsolve]` változó értékét, hogy több információt kapjunk a számítás menetéről.

```
> infolevel[dsolve] := 5:
> alias( y=y ): # unalias y
```

A változatosság kedvéért a differenciálegyenletet nem a **diff**-fel, hanem a **D** operátorral adjuk meg.

```
> ODE := ( y * D(y)= sqrt( y^2 - 2*y + 1) )(x);
```

$$ODE := \mathrm{y}(x)\,\mathrm{D}(y)(x) = \sqrt{(-1 + \mathrm{y}(x))^2}$$

A **dsolve** megérti ezt a jelölést is.

```
> dsolve( ODE, y(x) );
```

```
dsolve/diffeq/dsol1:  -> first order, first degree
methods :
dsolve/diffeq/dsol1:  trying linear bernoulli
dsolve/diffeq/dsol1:  trying separable
dsolve/diffeq/sepsol:  solving separable d.e.
dsolve/diffeq/dsol1:  separable successful
```

$$\frac{y(x)}{\sqrt{y(x)^2-2\,y(x)+1}}-\frac{y(x)^2}{\sqrt{y(x)^2-2\,y(x)+1}}+\frac{\ln(-1+y(x))}{\sqrt{y(x)^2-2\,y(x)+1}}$$
$$-\frac{\ln(-1+y(x))\,y(x)}{\sqrt{y(x)^2-2\,y(x)+1}}+x=_C1$$

az eredmény további egyszerűsítéssel a következő alakra hozható:

```
> simplify(");
```

$$-y(x)\operatorname{csgn}(-1+y(x))+x-\operatorname{csgn}(-1+y(x))\ln(-1+y(x))=_C1$$

A Maple láthatóan ügyel arra, hogy melyek az érvényes egyszerűsítések. Ezért ad vissza az explicit módszer két (lehetséges) megoldást.

```
> infolevel[dsolve] := 0:  # restore information level
> dsolve( ODE, y(x), explicit = true );
```

$$y(x)=e^{(-\operatorname{LambertW}(e^{(-1-x+_C1)})-1-x+_C1)}+1,$$
$$y(x)=e^{(-\operatorname{LambertW}(e^{(-1+x-_C1)})-1+x-_C1)}+1$$

A differenciálegyenlet tartalmazhat több változót is. Tekintsük például azt a differenciálegyenletet, amely az x tengely mentén pozitív irányban haladó ember által a hosszúságú kötéllel maga után vontatott objektum pályáját írja le:

$$y'=-\frac{y}{\sqrt{a^2-y^2}}$$

```
> alias( y=y(x) ):
> ODE := diff(y,x) = -y / sqrt(a^2-y^2);
```

$$ODE := \frac{\partial}{\partial x}y=-\frac{y}{\sqrt{a^2-y^2}}$$

```
> solution := dsolve( ODE, y );
```

$$solution := \sqrt{a^2-y^2}-\frac{a^2\operatorname{arctanh}(\frac{a^2}{\sqrt{a^2}\sqrt{a^2-y^2}})}{\sqrt{a^2}}+x=_C1$$

Sokkal egyszerűbb a megoldás, ha föltesszük, hogy a pozitív.

```
> solution := simplify( ", assume=positive );
```

$$solution := \sqrt{a^2-y^2}-a\operatorname{arctanh}(\frac{a}{\sqrt{a^2-y^2}})+x=_C1$$

Figyeljük meg, hogy a Maple nem veszi észre, hogy a 0 konstans függvény is megoldása a differenciálegyenletnek.

Ha az ember az origóból indul el, s az objektum ekkor a $(0, a)$ helyen van, vagyis ha a kezdeti föltétel $y(0) = a$, megpróbálhatjuk a `_C1` konstanst meghatározni:

```
> subs( {x=0,y=a}, solution );
```

```
Error, division by zero
```

Mi a hiba? A Maple a megoldást olyan implicit formulával írta föl, amely a kezdeti érték behelyettesítésekor hibás eredményre vezetett. Ha rájövünk, hogy $\operatorname{arctanh}(1/x)$ és $\operatorname{arctanh}(x)$ csupán egy konstansban tér el, próbálkozhatunk a következő implicit megoldással is:

```
> solution := sqrt(a^2-y^2) -
>    a*arctanh( sqrt(a^2-y^2)/a ) + x = C;
```

$$solution := \sqrt{a^2 - y^2} - a \operatorname{arctanh}(\frac{\sqrt{a^2 - y^2}}{a}) + x = C$$

Ellenőrizzük, hogy ez tényleg megoldás.

```
> diff( solution, x );
```

$$-\frac{y\,(\frac{\partial}{\partial x}y)}{\sqrt{a^2 - y^2}} + \frac{y\,(\frac{\partial}{\partial x}y)}{\sqrt{a^2 - y^2}\,(1 - \frac{a^2 - y^2}{a^2})} + 1 = 0$$

```
> Diff(y,x) = solve( ", diff(y,x) );
```

$$\frac{\partial}{\partial x}y = -\frac{y}{\sqrt{a^2 - y^2}}$$

Most már gond nélkül elvégezhető a behelyettesítés.

```
> (eval@subs)( {x=0,y=a}, solution );
```

$$0 = C$$

Csábítónak tűnhet ez a helyettesítés is:

```
> subs( x=0, solution );
```

$$\sqrt{a^2 - \mathrm{y}(0)^2} - a \operatorname{arctanh}(\frac{\sqrt{a^2 - \mathrm{y}(0)^2}}{a}) = C$$

```
> subs( y(0)=0, " );
```

$$\sqrt{a^2 - \mathrm{y}(0)^2} - a \operatorname{arctanh}(\frac{\sqrt{a^2 - \mathrm{y}(0)^2}}{a}) = C$$

Azért nem a várt eredményt kaptuk, mert korábban `y`-t az `y(x)` álneveként vezettük be. Az `x=0` első helyettesítés után visszaadott `y(0)` objektum mást jelent, mint ha egyszerűen csak `y(0)`-t gépelünk be.

```
> subs( subs(x=0,y) = a, " );
```

$$-a \operatorname{arctanh}(0) = C$$

```
> subs( op(0,y)(0) = a, "" );
```

$$-a \operatorname{arctanh}(0) = C$$

Az eredetileg követett módszerhez képest ezek elég trükkös megoldások. A példát a „tractrix" néven ismert megoldásgörbe leírásával zárjuk (v. ö. [69]).

```
> Tractrix := subs( C=0, solution );
```

$$Tractrix := \sqrt{a^2 - y^2} - a\operatorname{arctanh}(\frac{\sqrt{a^2 - y^2}}{a}) + x = 0$$

A Maple egyébként lehetővé teszi kezdeti- vagy peremérték problémák közvetlen megadását is. Ezt a matematikai inga

$$u'' + \omega^2 u = 0$$

differenciálegyenletével illusztráljuk, amelyben u a vízszintes kitérést jelenti. (Lásd 17.3. ábra.) Az álnevek használata az előző példa kapcsán leírtakhoz hasonló bonyodalmakat okozhatna, ezért inkább lemondunk róluk.

```
> ODE := diff(u(t),[t$2]) + omega^2*u(t) = 0;
```

$$ODE := (\frac{\partial^2}{\partial t^2}\operatorname{u}(t)) + \omega^2 \operatorname{u}(t) = 0$$

```
> dsolve( { ODE, u(0)=2, D(u)(0)=3 }, u(t) );
```

$$\operatorname{u}(t) = 2\cos(\omega\, t) + 3\,\frac{\sin(\omega\, t)}{\omega}$$

Az első deriváltat a 0 helyen `D(u)(0)` alakban adhatjuk meg. A további $u''(0)$, $u'''(0), \ldots$, magasabb rendű deriváltak alakja `(D@@(u))(0)`, `(D@@@(u))(0)`, ...

```
> dsolve( { ODE, u(0)=1, u(2)=3 }, u(t) );
```

$$\operatorname{u}(t) = \cos(\omega\, t) - \frac{(\cos(\omega)^2 - 2)\sin(\omega\, t)}{\sin(\omega)\cos(\omega)}$$

```
> combine(");
```

$$\operatorname{u}(t) = \frac{-\sin(\omega\, t - 2\,\omega) + 3\sin(\omega\, t)}{\sin(2\,\omega)}$$

A Maple-ben egyszerűen elérhető a differenciálegyenletek megoldására használható *integráltranszformációs módszer.* Példaként a csillapított lineáris oszcillátor válaszfüggvényére alkalmazott Laplace-transzformációt tekintjük. A megfelelő kezdetiérték probléma:

```
> ODE := diff(u(t),[t$2]) + 2*d*omega*diff(u(t),t)
>   + omega^2*u(t) = Heaviside(t);
```

$$ODE := (\frac{\partial^2}{\partial t^2}\operatorname{u}(t)) + 2\,d\,\omega\,(\frac{\partial}{\partial t}\operatorname{u}(t)) + \omega^2\operatorname{u}(t) = \operatorname{Heaviside}(t)$$

```
> initvals := u(0)=0, D(u)(0)=0:
> solution := dsolve( { ODE, initvals }, u(t),
>   method = laplace );
```

$$solution := \mathrm{u}(t) = \frac{1}{\omega^2} + \frac{e^{(-d\,\omega\,t)}\cos(\sqrt{-\omega^2\,(d-1)\,(d+1)}\,t)}{\omega^2\,(d-1)\,(d+1)}$$
$$-\frac{e^{(-d\,\omega\,t)}\,d^2\cos(\sqrt{-\omega^2\,(d-1)\,(d+1)}\,t)}{\omega^2\,(d-1)\,(d+1)}$$
$$+\frac{e^{(-d\,\omega\,t)}\sqrt{\omega^2 - d^2\,\omega^2}\,d\sin(\sqrt{-\omega^2\,(d-1)\,(d+1)}\,t)}{\omega^3\,(d-1)\,(d+1)}$$

```
> simplify(");
```

$$\mathrm{u}(t) = -\Big(\omega - \omega\,d^2 - e^{(-d\,\omega\,t)}\cos(\sqrt{-\omega^2\,(-1+d^2)}\,t)\,\omega$$
$$+\,e^{(-d\,\omega\,t)}\,d^2\cos(\sqrt{-\omega^2\,(-1+d^2)}\,t)\,\omega$$
$$-\,e^{(-d\,\omega\,t)}\sqrt{-\omega^2\,(-1+d^2)}\,d\sin(\sqrt{-\omega^2\,(-1+d^2)}\,t)\Big)\Big/\Big(\omega^3$$
$$(-1+d^2)\Big)$$

```
> solution := collect( ", {sin,cos}, simplify );
```

$$solution := \mathrm{u}(t) = -\frac{e^{(-d\,\omega\,t)}\cos(\sqrt{-\omega^2\,(-1+d^2)}\,t)}{\omega^2}$$
$$+\frac{e^{(-d\,\omega\,t)}\sqrt{-\omega^2\,(-1+d^2)}\,d\sin(\sqrt{-\omega^2\,(-1+d^2)}\,t)}{\omega^3\,(-1+d^2)} + \frac{1}{\omega^2}$$

A Maple nem tesz különbséget a csillapítás különböző esetei között ($d = 0$ csillapítás nélküli, $0 < d < 1$ alulvezérelt, $d = 1$ kritikus, $d < 1$ túlvezérelt). A legtöbb esetben a fönti megoldás az éppen legkényelmesebb formára egyszerűsíthető, ha explicit föltételeket teszünk, vagy közvetlenül megadjuk a paraméterek értékét.

```
> d := 0:
> simplify( solution, assume=positive ); # no damping
```

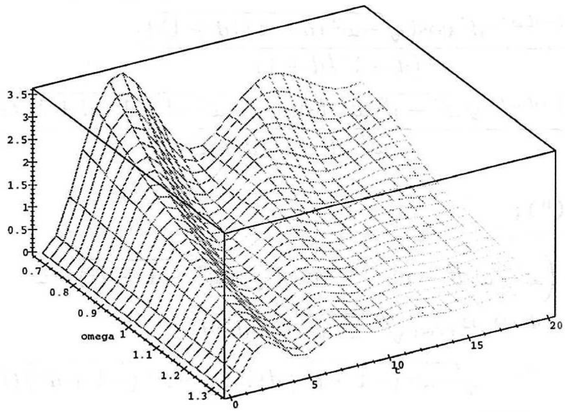
$$\mathrm{u}(t) = -\frac{\cos(\omega\,t) - 1}{\omega^2}$$

```
> d := 1/6:
> simplify( solution, assume=positive ); # underdamping
```

$$\mathrm{u}(t) = -\frac{1}{35}\Big(35\,e^{(-1/6\,\omega\,t)}\cos\big(\frac{1}{6}\sqrt{35}\,\omega\,t\big) +$$
$$e^{(-1/6\,\omega\,t)}\sqrt{35}\sin\big(\frac{1}{6}\sqrt{35}\,\omega\,t\big) - 35\Big)\Big/\,\omega^2$$

Ábrázoljuk ezt a megoldást különböző frekvencia értékekre. (Lásd 17.1. ábra.)

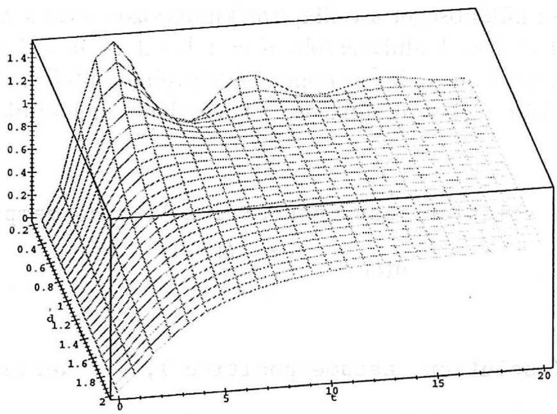
```
> plot3d( rhs("), omega=2/3..4/3, t=0..20,
>   style=hidden, orientation=[-30,45], axes=BOXED );
```



17.1. ábra: Az alulvezérelt oszcillátor válaszfüggvénye

Rögzített frekvencia esetén a különböző d értékekhez tartozó megoldásokat hasonlóan ábrázolhatjuk. (Lásd 17.2. ábra.)

```
> d := 'd': omega := 1:
> plot3d( rhs(solution), d=1/5..2, t=0..20,
>   style=hidden, orientation=[-10,45], axes=BOXED );
```



17.2. ábra: Az oszcillátor különböző csillapítási tényezőkhöz tartozó válaszfüggvénye

Az ábra alapján világos a csillapítás és a rendszer válasza közti kapcsolat: a kis d értékeknek megfelelő alulvezérléssel „túllövünk a célon", nagy d értékek mellett meg lassan válaszol a rendszer az input jelekre.

A többi integráltranszformáció (`fourier`, `fouriersin`, `fouriercos`, `hilbert` és `hankel`) szintén alkalmazható differenciálegyenletek megoldására a Maple-ben. Még egy példa:

```
> diff(u(t),[t$2]) + 3*diff(u(t),t)+ 2*u(t)=exp(-abs(t));
```

$$(\frac{\partial^2}{\partial t^2}\,\mathrm{u}(t)) + 3\,(\frac{\partial}{\partial t}\,\mathrm{u}(t)) + 2\,\mathrm{u}(t) = e^{(-|t|)}$$

```
> dsolve( ", u(t), method = fourier );
```

$$\mathrm{u}(t) = \frac{1}{6}\,e^t\,\mathrm{Heaviside}(-t) + \frac{2}{3}\,e^{(-2\,t)}\,\mathrm{Heaviside}(t) + e^{(-t)}\,t\,\mathrm{Heaviside}(t) - \frac{1}{2}\,e^{(-t)}\,\mathrm{Heaviside}(t)$$

A **dsolve** lineáris differenciálegyenletekből álló rendszert is meg tud oldani.

```
> ODE := diff(f(t),[t$2]) - 6*diff(g(t),t) =6*sin(t),
>   6*diff(g(t),[t$2]) + c^2*diff(f(t),t) = 6*cos(t);
```

$$ODE := (\frac{\partial^2}{\partial t^2}\,\mathrm{f}(t)) - 6\,(\frac{\partial}{\partial t}\,\mathrm{g}(t)) = 6\sin(t),\ 6\,(\frac{\partial^2}{\partial t^2}\,\mathrm{g}(t)) + c^2\,(\frac{\partial}{\partial t}\,\mathrm{f}(t)) = 6\cos(t)$$

```
> initvals := f(0)= 0, g(0)=1, D(f)(0)=0, D(g)(0)=1:
> funcs := {f(t),g(t)}:
> dsolve( { ODE, initvals }, funcs, method = laplace );
```

$$\left\{\begin{aligned}\mathrm{f}(t) &= 12\,\frac{\sin(t)}{(c-1)\,(c+1)} + \frac{6}{c^2} - 12\,\frac{\sin(c\,t)}{(c-1)\,(c+1)\,c} + 6\,\frac{\cos(c\,t)}{(c-1)\,(c+1)\,c^2} - 6\,\frac{\cos(c\,t)}{(c-1)\,(c+1)},\\ \mathrm{g}(t) &= \frac{\cos(t)}{(c-1)\,(c+1)} + \frac{\cos(t)\,c^2}{(c-1)\,(c+1)} - 2\,\frac{\cos(c\,t)}{(c-1)\,(c+1)} - \frac{\sin(c\,t)}{(c-1)\,(c+1)\,c} + \frac{c\sin(c\,t)}{(c-1)\,(c+1)}\end{aligned}\right\}$$

```
> collect( ", {cos(c*t),sin(c*t)}, normal );
```

$$\left\{\mathrm{f}(t) = -6\,\frac{\cos(c\,t)}{c^2} - 12\,\frac{\sin(c\,t)}{(c-1)\,(c+1)\,c} + 6\,\frac{2\sin(t)\,c^2 + c^2 - 1}{(c-1)\,(c+1)\,c^2},\ \mathrm{g}(t) = -2\,\frac{\cos(c\,t)}{(c-1)\,(c+1)} + \frac{\sin(c\,t)}{c} + \frac{(1+c^2)\cos(t)}{(c-1)\,(c+1)}\right\}$$

A 17.1. és a 17.2. táblázatban fölsoroltuk a közönséges differenciálegyenletek azon típusait, melyeket a **dsolve** jelenlegi változata meg tud oldani.

A KDE típusa	**A KDE alakja**
lineáris	$y' + P(x)y = Q(x)$
egzakt	$y' = -\dfrac{P(x,y)}{Q(x,y)}$ és $\dfrac{\partial Q}{\partial x} = \dfrac{\partial P}{\partial y}$
inegzakt	$F(x,y)y' + G(x,y)$, ha található hozzá integráló tényező
szeparálható	$y' = f(x)g(y)$
homogén	$y' = F(xy^n)\dfrac{y}{x}$
magasabb rendű	$x = F((y,y')$
Bernoulli-féle	$y' + P(x)y = Q(x)y^n$
Clairaut-féle	$y = xy' + F(y')$
Riccati-féle	$y' = P(x)y^2 + Q(x)y' + R(x)$

17.1. táblázat: Elsőrendű megoldható KDE típusok

A KDE típusa	**A KDE alakja**
konstans eüh. lineáris	$ay'' + by' + cy = d(x)$, ahol $a,b,c \in \mathbb{C}$
Euler-féle	$x^2y'' + axy' + by = c(x)$
Bessel-féle	$x^2y'' + (2k+1)xy' + (\alpha^2x^{2r} + \beta^2)y = 0$, ahol $k,\alpha,\beta,r \in \mathbb{C}$ és $\alpha r \neq 0$
hipergeometrikus	$x(1-x)y'' + \big((c-(a+b+1)x\big)y' - aby = 0$, ahol $a,b,c \in \mathbb{C}$
Riemann–Papperitz	$y'' + Py' + Qy = 0$, ahol $P = \frac{1-\alpha_1-\beta_1}{x-a} + \frac{1-\alpha_2-\beta_2}{x-b} + \frac{1-\alpha_3-\beta_3}{x-c}$, $Q = \left(\frac{\alpha_1\beta_1(a-b)(a-c)}{x-a} + \frac{\alpha_2\beta_2(b-a)(b-c)}{x-b} + \frac{\alpha_3\beta_3(c-a)(c-b)}{x-c}\right) \times \frac{1}{(x-a)(x-b)(x-c)}$

17.2. táblázat: Másodrendű megoldható KDE típusok

Ezen kívül a Kovacic-algoritmust (lásd [121]) is implementálták a Maple-ben. Ennek segítségével olyan

$$P(x)y'' + Q(x)y' + R(x)y = 0$$

alakú homogén lineáris másodrendű KDE-ket próbálunk megoldani, ahol $P(x)$, $Q(x)$ és $R(x)$ racionális függvények. A módszer a Risch-algoritmushoz hasonló döntési eljárás: adott lineáris másodrendű KDE-re a Kovacic-algoritmus eldönti, hogy létezik-e zárt alakban fölírható Liouville-megoldás; ha a válasz igen,

meg is ad egy ilyen megoldást. Magasabb rendű lineáris differenciálegyenletekre két fontos döntési eljárást implementáltak: az Abromov-féle RATLODE-t (RATional Linear ODE, v. ö. [2]), amely azt dönti el, hogy az ODE együtthatóit tartalmazó tartományban van-e zárt alakban fölírható megoldás, továbbá Bronstein EXPLODE (EXPonential Linear ODE, lásd [22]) algoritmusát, amely $\exp(\int f(x)\,dx)$ alakú megoldásokat határoz meg. A Maple-ben implementált módszerekről kellemes bevezetőt találunk a [123] cikkben. Az EXPLODE-t használja föl a következő példa:

```
> alias( y=y(x) ):
> x^2*diff(y,[x$3]) + (x-3*x^2)*diff(y,[x$2]) +
>    (4*x^2-2*x)*diff(y,x) + (x-2*x^2)*y = 0;
```

$$x^2\left(\frac{\partial^3}{\partial x^3}y\right)+(x-3x^2)\left(\frac{\partial^2}{\partial x^2}y\right)+(4x^2-2x)\left(\frac{\partial}{\partial x}y\right)+(x-2x^2)\,y=0$$

```
> dsolve( ", y );
```

$$\begin{aligned} y = {}&_C1\,e^x+_C2(-2\,x\,e^x\,\mathrm{BesselY}(0,x)\\ &-x\,e^x\,\pi\,\mathrm{StruveH}(0,x)\,\mathrm{BesselY}(1,x)\\ &+x\,e^x\,\pi\,\mathrm{StruveH}(1,x)\,\mathrm{BesselY}(0,x))+_C3(-2\,x\,e^x\,\mathrm{BesselJ}(0,x)\\ &-x\,e^x\,\pi\,\mathrm{StruveH}(0,x)\,\mathrm{BesselJ}(1,x)\\ &+x\,e^x\,\pi\,\mathrm{StruveH}(1,x)\,\mathrm{BesselJ}(0,x)) \end{aligned}$$

Nézzünk egy Bessel-féle egyenletet is:

```
> alias( y=y(x) ):
> diff(y,[x$2]) + 3*diff(y,x)/x +(x^2-15)/x^2*y - 1/x;
```

$$\left(\frac{\partial^2}{\partial x^2}y\right)+3\,\frac{\frac{\partial}{\partial x}y}{x}+\frac{(x^2-15)\,y}{x^2}-\frac{1}{x}$$

```
> dsolve( ", y );
```

$$y=\frac{192+16\,x^2+x^4}{x^5}+\frac{_C1\,\mathrm{BesselJ}(4,x)}{x}+\frac{_C2\,\mathrm{BesselY}(4,x)}{x}$$

Alternatív output formátumot kínál a *bázismegoldások* fölhasználása.

```
> dsolve( "", y, output = basis );
```

$$\left[\left[\frac{\mathrm{BesselJ}(4,x)}{x},\frac{\mathrm{BesselY}(4,x)}{x}\right],\frac{192+16\,x^2+x^4}{x^5}\right]$$

Néha még akkor is segítségre szorul a Maple, ha sikeresen fölismerte a KDE típusát:

```
> infolevel[dsolve] := 2:
> ODE := diff(y(x),x) + 2*y(x)*exp(x) - y(x)^2
>    - exp(2*x) - exp(x) = 0;
```

$$ODE:=\left(\frac{\partial}{\partial x}\mathrm{y}(x)\right)+2\,\mathrm{y}(x)\,e^x-\mathrm{y}(x)^2-e^{(2\,x)}-e^x=0$$

```
> dsolve( ODE, y(x) );

dsolve/diffeq/dsol1:
-> first order, first degree methods :
dsolve:   Warning: no solutions found
```

Könnyen ellenőrizhető, hogy e^x megoldás.

```
> subs( y(x) = exp(x), ODE ): expand(");
```

$$0 = 0$$

A keresett $y(x)$ függvényt írjuk föl $z(x) + e^x$ alakban, és próbálkozzunk újra.

```
> subs( y(x) = z(x) + exp(x), ODE ): expand(");
```

$$(\frac{\partial}{\partial x} \mathrm{z}(x)) - \mathrm{z}(x)^2 = 0$$

```
> dsolve( ", z(x), explicit = true );

dsolve/diffeq/dsol1:
-> first order, first degree
methods :
dsolve/diffeq/bernsol:  trying Bernoulli solution
dsolve/diffeq/linearsol:  solving 1st order linear d.e.
dsolve/diffeq/dsol1:  linear bernoulli successful
```

$$\mathrm{z}(x) = \frac{1}{-x + _C1}$$

Vizsgáljuk meg az általános megoldást:

```
> solution := exp(x) + rhs(");
```

$$solution := e^x + \frac{1}{-x + _C1}$$

```
> subs( y(x) = solution, ODE ): testeq(");
```

$$true$$

A Maple néha csak részben oldja meg a feladatot:

```
> alias( y=y(x) ):
> ODE := diff(y,[x$3]) + (2+2*x)*diff(y,[x$2]) +
>   (4*x+4-1/x)*diff(y,x) + 2*y = 0;
```

$$ODE := (\frac{\partial^3}{\partial x^3} y) + (2 + 2\,x)\,(\frac{\partial^2}{\partial x^2} y) + (4\,x + 4 - \frac{1}{x})\,(\frac{\partial}{\partial x} y) + 2\,y = 0$$

```
> dsolve( ODE, y );
```

$$y = _C1\, e^{(-x^2)} + e^{(-x^2)} \int \mathrm{DESol}\Big(\Big\{x\,(\frac{\partial^2}{\partial x^2} _\mathrm{Y}(x)) + (2\,x - 4\,x^2)\,(\frac{\partial}{\partial x} _\mathrm{Y}(x)) + (-4\,x^2 - 2\,x - 1 + 4\,x^3)\,_\mathrm{Y}(x)\Big\}, \Big\{_\mathrm{Y}(x)\Big\}\Big) dx$$

Ez a **RootOf** jelöléssel analóg: **DESol** a differenciálegyenlet valamelyik megoldását reprezentálja, amelyet további földolgozásra átadhatunk a **diff**, **series**, **evalf** stb. eljárásoknak. A **?DESol** paranccsal még többet tudhatunk meg erről a lehetőségről. A fönti megoldást tehát így ellenőrizhetjük:

```
> simplify( subs(",ODE) );
```

$$0 = 0$$

A 17.3. táblázat a **dsolve** azon opcióit tartalmazza, amelyeket akkor alkalmazhatunk, ha az analitikus megoldás előállítását kértük a `type=exact` opcióval.

Opció	**Jelentése**
`method = fourier`	Fourier módszer
`method = fouriercos`	a Fourier-féle koszinusz transzformált módszer
`method = fouriersin`	a Fourier-féle szinusz transzformált módszer
`method = hankel`	a Hankel transzformált módszer
`method = hilbert`	a Hibert transzformált módszer
`method = laplace`	a Laplace transzformált módszer
`method = matrixexp`	a mátrix exponenciális módszer
`output = basis`	bázismegoldás és partikuláris megoldás generálása
`explicit = true`	explicit megoldás keresése
`explicit = false`	implicit megoldás

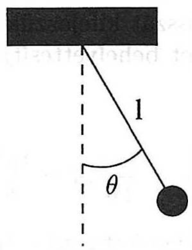
17.3. táblázat: A **dsolve**(*ODE*, `type=exact`) kiegészítő opciói

17.3. Taylor-sor módszer

Ha nem tudtuk meghatározni a KDE analitikus megoldását, még mindig megpróbálhatjuk kiszámoltatni a Maple-lel Taylor-sor segítségével való közelítését. Ezt a módszert alkalmazzuk a 17.3. ábrán látható ingára, melynek mozgását az

$$l\,\theta'' = -g\sin\theta$$

differenciálegyenlet írja le (l az inga hossza, g a nehézségi gyorsulás, θ pedig a fonálnak a függőlegessel bezárt szöge).



17.3. ábra: Inga

```
> ODE := l*diff(theta(t),[t$2]) =-g*sin(theta(t));
```

$$ODE := l\,(\frac{\partial^2}{\partial t^2}\,\theta(t)) = -g\sin(\theta(t))$$

```
> initvals := theta(0)=0, D(theta)(0)=v[0]/l:
> Order := 9:
> solution := dsolve( { ODE, initvals }, theta(t),
>   type = series );
```

$$solution := \theta(t) = \frac{v_0\,t}{l} - \frac{1}{6}\,\frac{g\,v_0\,t^3}{l^2} + \frac{1}{120}\,\frac{g\,v_0\,(g\,l + v_0{}^2)\,t^5}{l^4} - \frac{1}{5040}\,\frac{g\,v_0\,(g^2\,l^2 + 11\,g\,l\,v_0{}^2 + v_0{}^4)\,t^7}{l^6}$$

Az alábbiakban dimenzió nélküli változókkal írjuk föl ezt a megoldást. Először még ellenőrizzük a dimenziókat:

```
> subs( { v[0]=L/T, g=L/T^2, l=L }, " );
```

$$\theta(t) = \frac{t}{T} - \frac{1}{6}\,\frac{t^3}{T^3} + \frac{1}{60}\,\frac{t^5}{T^5} - \frac{13}{5040}\,\frac{t^7}{T^7}$$

Ezután a dimenzió nélküli a és b paramétereket határozzuk meg:

```
> nondimensional := v[0]^ev[0] * g^eg * l^el * t^et;
```

$$nondimensional := v_0{}^{ev_0}\,g^{eg}\,l^{el}\,t^{et}$$

```
> simplify( subs( { v[0]=L/T, g=L/T^2, l=L, t=T },
>   nondimensional ), assume=positive );
```

$$L^{(ev_0+eg+el)}\,T^{(-ev_0-2\,eg+et)}$$

```
> isolve( { seq( op(2,i)=0, i=" ) } );
```

$$\{et = -_N1 + _N2,\ ev_0 = -_N1 - _N2,\ el = _N1,\ eg = _N2\}$$

```
> assign("):
> { a = subs( _N1=-1, _N2=1, nondimensional),
>   b = subs( _N1=-1, _N2=0, nondimensional) };
```

$$\left\{a = \frac{g\,t^2}{l},\ b = \frac{v_0\,t}{l}\right\}$$

A g gravitációt és a fonál l hosszát kifejezzük a dimenzió nélküli változókkal, majd az így kapott kifejezéseket behelyettesítjük a differenciálegyenlet megoldásába:

```
> solve( ", {g,l} );
```

$$\left\{g = \frac{a\,v_0}{t\,b},\ l = \frac{v_0\,t}{b}\right\}$$

```
> subs( ", convert( rhs(solution), 'polynom' ) );
```

$$b - \frac{1}{6}\,a\,b + \frac{1}{120}\,\frac{a\,b^3\left(\frac{a\,v_0{}^2}{b^2} + v_0{}^2\right)}{v_0{}^2} - \frac{1}{5040}\,\frac{a\,b^5\,(\frac{a^2\,v_0{}^4}{b^4} + 11\,\frac{a\,v_0{}^4}{b^2} + v_0{}^4)}{v_0{}^4}$$

```
> map( normal, " );
```

$$b - \frac{1}{6}\,a\,b + \frac{1}{120}\,a\,b\,(a + b^2) - \frac{1}{5040}\,a\,b\,(a^2 + 11\,a\,b^2 + b^4)$$

Hasonlítsuk össze az eredményt a $\sin t$ sorfejtésével.

```
> series( sin(t), t, 9 );
```

$$t - \frac{1}{6}t^3 + \frac{1}{120}t^5 - \frac{1}{5040}t^7 + O(t^9)$$

17.4. A hatványsor módszer

Az olyan modern számítógépes algebrai rendszerekből, mint a Maple, természetesen nem hiányozhat a differenciálegyenletek megoldására szolgáló hatványsor módszer sem. De jegyezzük meg, hogy ez a módszer csak polinomegyütthatós lineáris differenciálegyenletekre működik. Első példánk egy Bessel-féle differenciálegyenlet:

$$xy'' + y' + 4x^2y = 0.$$

A Maple megtalálja a pontos megoldást.

```
> eqn := x*diff(y(x),[x$2]) + diff(y(x),x) +
>   4*x^2*y(x) = 0;
```

$$eqn := x\left(\frac{\partial^2}{\partial x^2}\,y(x)\right) + \left(\frac{\partial}{\partial x}\,y(x)\right) + 4\,x^2\,y(x) = 0$$

```
> dsolve( eqn, y(x) );
```

$$y(x) = _C1\,\mathrm{BesselY}(0, \frac{4}{3}x^{3/2}) + _C2\,\mathrm{BesselJ}(0, \frac{4}{3}x^{3/2})$$

Az alábbiakban az $y(0) = 1, y'(0) = 0$ kezdeti értékeknek megfelelő megoldást hatványsorként állítjuk elő.

```
> initvals := y(0)=1, D(y)(0)=0:
> with( powseries ):  # load the power series package
> solution := powsolve( { eqn, initvals } );
```

$$solution := \mathbf{proc}(powparm)\ \ldots\ \mathbf{end}$$

```
> tpsform( solution, x, 15 ); # truncated powerseries
```

$$1 - \frac{4}{9}x^3 + \frac{4}{81}x^6 - \frac{16}{6561}x^9 + \frac{4}{59049}x^{12} + O(x^{15})$$

A Maple az együtthatókra vonatkozó rekurzív képletet is meg tudja adni.

```
> solution(_k);
```

$$-4\,\frac{a(_k-3)}{_k^2}$$

Emlékeztetünk arra, hogy az aláhúzással kezdődő neveket maga a Maple használja, a legutolsó képlet tehát az

$$a_k = -4\frac{a_{k-3}}{k^2}$$

rekurzív relációnak felel meg.

Második klasszikus példánk a kvantummechanikából származik: oldjuk meg az egydimenziós harmonikus oszcillátor egyenletét. Dimenzió nélküli egységekkel a Schrödinger egyenlet a következő alakú lesz:

$$\frac{d^2y(x)}{dx^2} + (\lambda - x^2)y(x) = 0.$$

Az egyenlet levezetése megtalálható a 17.6 alfejezetben.

```
> alias( y=y(x), h=h(x) ):
> eqn := diff(y,[x$2]) + (lambda-x^2)*y = 0
```

$$eqn := \left(\frac{\partial^2}{\partial x^2}\, y\right) + (\lambda - x^2)\, y = 0$$

Az aszimptotikus viselkedés analízise alapján az $y(x) = h(x)e^{-x^2/2}$ helyettesítés látszik célszerűnek. A h-ra vonatkozó differenciálegyenlet:

```
> subs( y = exp(-x^2/2)*h, " ):
> collect( ", exp(-x^2/2) ) / exp(-x^2/2);
```

$$-h + x^2\, h - 2\, x \left(\frac{\partial}{\partial x}\, h\right) + \left(\frac{\partial^2}{\partial x^2}\, h\right) + (\lambda - x^2)\, h = 0$$

```
> eqn := collect( ", [diff(h,[x$2]), diff(h,x), h] );
```

$$eqn := \left(\frac{\partial^2}{\partial x^2}\, h\right) - 2\, x \left(\frac{\partial}{\partial x}\, h\right) + (-1 + \lambda)\, h = 0$$

A differenciálegyenletet a hatványsor módszerrel oldjuk meg:

```
> with(powseries):
> H := powsolve(eqn):
> h := tpsform(H,x,10); # a few terms
```

$$h := C0 + C1\, x + \frac{1}{2}\, C0\, (1 - \lambda)\, x^2 + \frac{1}{6}\, C1\, (3 - \lambda)\, x^3 +$$
$$\frac{1}{24}\, C0\, (1 - \lambda)\, (5 - \lambda)\, x^4 + \frac{1}{120}\, C1\, (3 - \lambda)\, (7 - \lambda)\, x^5 +$$
$$\frac{1}{720}\, C0\, (1 - \lambda)\, (5 - \lambda)\, (9 - \lambda)\, x^6 +$$
$$\frac{1}{5040}\, C1\, (3 - \lambda)\, (7 - \lambda)\, (11 - \lambda) x^7 +$$
$$\frac{1}{40320}\, C0\, (1 - \lambda)\, (5 - \lambda)\, (9 - \lambda)\, (13 - \lambda)\, x^8 +$$
$$\frac{1}{362880}\, C1\, (3 - \lambda)\, (7 - \lambda)\, (11 - \lambda)\, (15 - \lambda)\, x^9 + O(x^{10})$$

```
> collect( convert( ", 'polynom' ), [C0,C1] );
```

$$\left(1+\frac{1}{2}(1-\lambda)x^2+\frac{1}{24}(1-\lambda)(5-\lambda)x^4+\frac{1}{720}(1-\lambda)(5-\lambda)(9-\lambda)x^6\right.$$
$$\left.+\frac{1}{40320}(1-\lambda)(5-\lambda)(9-\lambda)(13-\lambda)x^8\right)C0+$$
$$\left(x+\frac{1}{6}(3-\lambda)x^3+\frac{1}{120}(3-\lambda)(7-\lambda)x^5+\frac{1}{5040}(3-\lambda)(7-\lambda)(11-\lambda)x^7\right.$$
$$\left.+\frac{1}{362880}(3-\lambda)(7-\lambda)(11-\lambda)(15-\lambda)x^9\right)C1$$

Nézzük meg, hogy h sorának együtthatóira milyan rekurziót kapunk.

```
> H(_k);
```

$$\frac{\mathrm{a}(_k-2)(-3-\lambda+2_k)}{_k(_k-1)}$$

Ennek jelentése

$$a_k=-\frac{(3+\lambda-2k)a_{k-2}}{k(k-1)},$$

vagy ekvivalens módon

$$(k+1)(k+2)a_{k+2}=(2k+1-\lambda)a_k$$

tetszőleges k-ra. A sorfejtés akkor és csak akkor lesz véges, ha valamely egész k-ra teljesül a $\lambda=2k+1$ egyenlőség, ami a harmonikus oszcillátor energiaszintjeinek híres kvantifikációját adja. A hullámfüggvényre kapott egyik példa:

```
> C0 := 1: C1 := 0: lambda := 9:
> tpsform(H,x,10): convert(",'polynom');
```

$$1-4x^2+\frac{4}{3}x^4$$

Ez a megoldás a negyedik Hermite polinom konstansszorosa.

```
> orthopoly[H](4,x) / 12;
```

$$1-4x^2+\frac{4}{3}x^4$$

17.5. Numerikus módszerek

Kezdetiérték problémák numerikus megoldását a `type = numeric` opcióval kérhetjük a Maple-től. Alapföltételezés szerint ekkor az RKF45 algoritmusnak megfelelő Fehlberg-féle negyedrendű Runge–Kutta módszert alkalmaz a rendszer (lásd [61]), de más numerikus módszerek is rendelkezésünkre állnak. Elsőként az $y(0)=0$, $y'(0)=-0.1$ kezdeti föltételekkel kiegészített

$$y''-(1-y^2)y'+y=0$$

van der Pol egyenletre alkalmazzuk a Runge–Kutta módszert.

Differenciálegyenletek rendszerének numerikus megoldásához a **dsolve** eljárást a `type = numeric` opcióval kell meghívni.

```
> alias( y=y(t), y0=y(0), yp0=D(y)(0) ):
> eqn := diff(y,[t$2]) - (1-y^2)*diff(y,t) + y= 0;
```

$$eqn := (\frac{\partial^2}{\partial t^2} y) - (1 - y^2)(\frac{\partial}{\partial t} y) + y = 0$$

```
> initvals := y0=0, yp0=-0.1:
> F := dsolve( {eqn, initvals}, y, type = numeric );
```

$$F := \mathbf{proc}(rkf45_x) \ldots \mathbf{end}$$

A numerikus megoldó algoritmus olyan listát ad vissza, amely három egyenlőséget tartalmaz: az első a t független változó értékét adja meg, a második az y függő változó értékét ezen a helyen, a harmadik pedig pedig az y' deriváltat ugyanitt. Két példa következik ennek illusztrálására:

```
> F(0);
```

$$[t = 0, y = 0, \frac{\partial}{\partial t} y = -.1]$$

```
> F(1);
```

$$[t = 1, y = -.1447686096006437, \frac{\partial}{\partial t} y = -.1781040958088073]$$

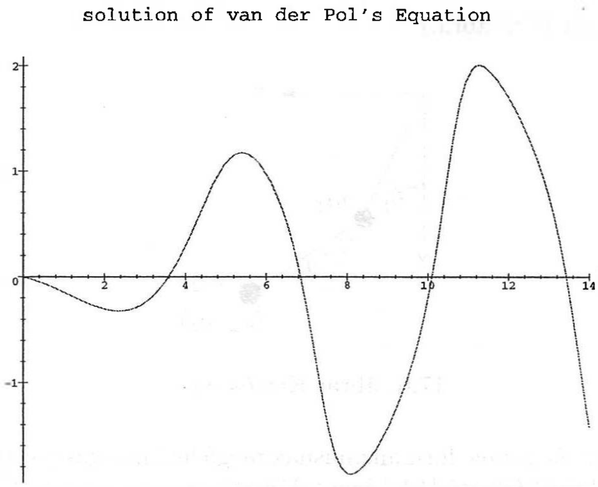
Jobban megértjük a megoldást , ha megpróbáljuk ábrázolni valamely számunkra érdekes tartományon. Először konvertáljuk függvénnyé a numerikus megoldást. Könnyen kiválasztható az y függő változó értékét definiáló egyenlőség jobb oldala.

```
> Y := t -> rhs( op(2,F(t)) );
```

$$Y := t \to \mathrm{rhs}(\mathrm{op}(2, \mathrm{F}(t)))$$

Ezután a **plot** parancsot használhatjuk a megoldás grafikonjának fölrajzolására. (Lásd 17.4. ábra.)

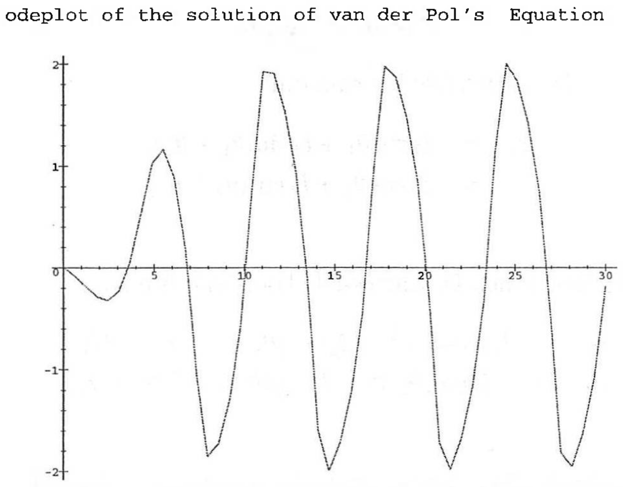
```
> plot( Y, 0..14, title=
>   `solution of van der Pol's Equation` );
```



17.4. ábra: A van der Pol egyenlet megoldásának grafikonja

Ha nincs szükségünk az adaptív rajzoló algoritmussal készült „szép” grafikonra, a megoldást ábrázolhatjuk a plots csomag **odeplot** eljárásával is.

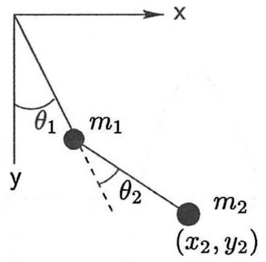
```
> plots[odeplot]( F, [t,y], 0..30, title=
>   'odeplot of the solution of van der Pol's Equation' );
```



17.5. ábra: A van der Pol egyenlet **odeplot**-tal fölrajzolt megoldása

Második, terjedelmesebb példánkban a numerikus KDE megoldó módszert olyan differenciálegyenletek rendszerére alkalmazzuk, amelyek a merev, surlódásmentes, két szabadsági fokú robotmanipulátor dinamikáját írják le (kettős inga). Az [58]-ban ismertetett dverk78 7,8 rendű folytonos Runge–Kutta módszert

használjuk. (Lásd 17.6. ábra.)



17.6. ábra: Kettős inga

Először az *Euler–Lagrange* formalizmusnak megfelelő mozgásegyenleteket vezetjük le. Az érdeklődő Olvasó [122]-ben találhatja meg a Newton–Euler formalizmus szerinti mozgásegyenletek levezetését. Legyen $\theta = (\theta_1, \theta_2)$ és $\dot{\theta} = (\dot{\theta}_1, \dot{\theta}_2)$, definiáljuk az

$$L(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta)$$

Lagrange-függvényt, amelyben $T(\theta, \dot{\theta})$ a mozgási, $V(\theta)$ a helyzeti energiát jelenti. A fönti konfiguráció esetében a mozgási energiát az m_1 tömeg T_1 és az m_2 tömeg T_2 mozgási energiájának összegeként kapjuk. Az m_1 tömeg mozgási energiája közvetlenül megadható:

$$T_1(\theta, \dot{\theta}) = \frac{1}{2} m_1 \dot{\theta}_1^2.$$

A végpont (x_2, y_2) Descartes-féle koordinátái

$$\begin{aligned} x_2 &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2), \\ y_2 &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2). \end{aligned}$$

A sebesség komponenseinek Descartes-féle koordinátái tehát

$$\begin{aligned} \dot{x}_2 &= l_1 (\cos\theta_1)\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2), \\ \dot{y}_2 &= -l_1 (\sin\theta_1)\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2). \end{aligned}$$

A sebesség abszolút értékének négyzetét véve a következőt kapjuk:

$$T_2(\theta, \dot{\theta}) = \frac{1}{2} m_2 \big(l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2 l_1 l_2 (\cos\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \big).$$

A helyzeti energiát a tömegpontok magassága határozza meg.

$$\begin{aligned} V_1(\theta) &= -m_1 g\, l_1 \cos\theta_1, \\ V_2(\theta) &= -m_2 g\, l_1 \cos\theta_1 - m_2 g\, l_2 (\cos\theta_1 + \theta_2). \end{aligned}$$

Az összes előző formulát megkaphattuk volna a Maple fölhasználásával.

```
> alias( theta[1]=theta[1](t), theta[2]=theta[2](t) ):
> macro( m1=m[1], m2=m[2], l1=l[1], l2=l[2],
> x1=x[1], x2=x[2], y1=y[1], y2 = y[2],
> t1=theta[1](t), t2=theta[2](t) ):
> x1 := l1*sin(t1): y1 := l1*cos(t1):
> x2 := l1*sin(t1) + l2*sin(t1+t2):
> y2 := l1*cos(t1) + l2*cos(t1+t2):
```

Az m_1 tömeg mozgási energiáját így számíthatjuk ki:

```
> T[1] := simplify( 1/2*m1*(diff(x1,t)^2 + diff(y1,t)^2) );
```

$$T_1 := \frac{1}{2} m_1 {l_1}^2 \left(\frac{\partial}{\partial t}\theta_1\right)^2$$

Az m_2 tömeg mozgási energiáját is hasonlóan kapjuk:

```
> T[2] := expand( 1/2*m2*(diff(x2,t)^2 + diff(y2,t)^2),
>   cos(t1+t2), sin(t1+t2) ):
> simplify(");
```

$$\begin{aligned}
&m_2\, l_1 \cos(\theta_1) \left(\frac{\partial}{\partial t}\theta_1\right)^2 l_2 \cos(\%1)\\
&\quad + m_2\, l_1 \cos(\theta_1) \left(\frac{\partial}{\partial t}\theta_1\right) l_2 \cos(\%1) \left(\frac{\partial}{\partial t}\theta_2\right) + \frac{1}{2} m_2 {l_1}^2 \left(\frac{\partial}{\partial t}\theta_1\right)^2\\
&\quad + m_2\, l_1 \sin(\theta_1) \left(\frac{\partial}{\partial t}\theta_1\right)^2 l_2 \sin(\%1)\\
&\quad + m_2\, l_1 \sin(\theta_1) \left(\frac{\partial}{\partial t}\theta_1\right) l_2 \sin(\%1) \left(\frac{\partial}{\partial t}\theta_2\right) + \frac{1}{2} m_2 {l_2}^2 \left(\frac{\partial}{\partial t}\theta_1\right)^2\\
&\quad + m_2 {l_2}^2 \left(\frac{\partial}{\partial t}\theta_1\right) \left(\frac{\partial}{\partial t}\theta_2\right) + \frac{1}{2} m_2 {l_2}^2 \left(\frac{\partial}{\partial t}\theta_2\right)^2\\
&\%1 := \theta_1 + \theta_2
\end{aligned}$$

További egyszerűsítésekre, így a véges Fourier sorok kiszámítására is szükség van.

```
> map( combine, collect(",diff(t1,t)), 'trig' ):
> T[2] := collect( ", [l1,l2,m2,cos(t2)], factor );
```

$$\begin{aligned}
T_2 := &\frac{1}{2} m_2 {l_1}^2 \left(\frac{\partial}{\partial t}\theta_1\right)^2 + \left(\frac{\partial}{\partial t}\theta_1\right) \left(\left(\frac{\partial}{\partial t}\theta_1\right) + \left(\frac{\partial}{\partial t}\theta_2\right)\right) \cos(\theta_2)\, m_2\, l_2\, l_1\\
&+ \frac{1}{2} \left(\left(\frac{\partial}{\partial t}\theta_1\right) + \left(\frac{\partial}{\partial t}\theta_2\right)\right)^2 m_2 {l_2}^2
\end{aligned}$$

Az Euler–Lagrange mozgásegyenletek a következők:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = 0, \qquad i = 1, 2.$$

Ezek fölhasználásával a következő vektoregyenletet írhatjuk föl:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = 0,$$

ahol $\ddot\theta = (\ddot\theta_1, \ddot\theta_2)$,

$$M(\theta) = \begin{pmatrix} m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 \end{pmatrix},$$

a Coriolis- vagy centripetális erőnek megfelelő $C(\theta, \dot\theta)$ tag

$$\begin{pmatrix} -m_2 l_1 l_2 \sin(\theta_2)\dot\theta_2(2\dot\theta_1 + \dot\theta_2) \\ m_2 l_1 l_2 \sin(\theta_2)\dot\theta_1^2 \end{pmatrix},$$

végül a gravitációnak megfelelő $G(\theta)$ tag definíciója

$$\begin{pmatrix} m_1\, g\, l_1 \sin\theta_1 + m_2\, g\, l_1 \sin\theta_1 + m_2\, g\, l_2 \sin(\theta_1 + \theta_2) \\ m_2\, g\, l_2 \sin(\theta_1 + \theta_2) \end{pmatrix}.$$

Lássuk, hogyan számítható ki ez a Maple segítségével. Először az m_1, illetve az m_2 tömeg helyzeti energiáját határozzuk meg.

```
> V[1] := - m1*g*l1*cos(t1):
> V[2] := - m2*g*l1*cos(t1) - m2*g*l2*cos(t1+t2):
> L := T[1]+T[2]-V[1]-V[2]: # Lagrangian
```

A $\dfrac{\partial L}{\partial \dot\theta_i}$ és a $\dfrac{\partial L}{\partial \theta_i}$ deriváltak közvetlenül nehezen számolhatók ki. Ezt a problémát segédváltozók bevezetésével oldjuk meg.

```
> L := subs( { t1=t_1, t2=t_2, diff(t1,t)=t1p,
>   diff(t2,t)=t2p }, L ):
> dL_dt1p := diff(L,t1p):
> dL_dt2p := diff(L,t2p):
> dL_dt1 := diff(L,t_1):
> dL_dt2 := diff(L,t_2):
> dL_dt1p := subs( { t_1=t1, t_2=t2, t1p=diff(t1,t),
>   t2p=diff(t2,t) }, dL_dt1p ):
> dL_dt2p := subs( { t_1=t1, t_2=t2, t1p=diff(t1,t),
>   t2p=diff(t2,t) }, dL_dt2p ):
> dL_dt1 := subs( { t_1=t1, t_2=t2, t1p=diff(t1,t),
>   t2p=diff(t2,t) }, dL_dt1 ):
> dL_dt2 := subs( { t_1=t1, t_2=t2, t1p=diff(t1,t),
>   t2p=diff(t2,t) }, dL_dt2 ):
```

Most már minden készen áll az Euler–Lagrange egyenleteken alapuló differenciálegyenletek definiálásához.

```
> ode[1] := collect( diff( dL_dt1p, t ) - dL_dt1 = 0,
> [diff(t1,[t$2]),diff(t2,[t$2]),diff(t1,t),dif f(t2,t)] );
```

$$ode_1 := (m_1\, l_1{}^2 + m_2\, l_1{}^2 + m_2\, l_2{}^2 + 2\cos(\theta_2)\, m_2\, l_2\, l_1)\,(\frac{\partial^2}{\partial t^2}\,\theta_1)$$

$$+ (m_2 l_2{}^2 + \cos(\theta_2)\, m_2 l_2 l_1) (\frac{\partial^2}{\partial t^2} \theta_2)$$
$$- 2 (\frac{\partial}{\partial t} \theta_1) \sin(\theta_2) (\frac{\partial}{\partial t} \theta_2)\, m_2 l_2 l_1 - (\frac{\partial}{\partial t} \theta_2)^2 \sin(\theta_2)\, m_2 l_2 l_1$$
$$+ m_2\, g\, l_2 \sin(\theta_1 + \theta_2) + m_1\, g\, l_1 \sin(\theta_1) + m_2\, g\, l_1 \sin(\theta_1) = 0$$

```
> ode[2] := collect( diff( dL_dt2p, t ) - dL_dt2 = 0,
> [diff(t1,[t$2]),diff(t2,[t$2]),diff(t1,t),diff(t2,t)] );
```

$$ode_2 := (m_2 l_2{}^2 + \cos(\theta_2)\, m_2 l_2 l_1) (\frac{\partial^2}{\partial t^2} \theta_1) + (\frac{\partial^2}{\partial t^2} \theta_2)\, m_2 l_2{}^2$$
$$+ \sin(\theta_2)\, m_2 l_2 l_1 (\frac{\partial}{\partial t} \theta_1)^2 + m_2\, g\, l_2 \sin(\theta_1 + \theta_2) = 0$$

Az egyszerűség kedvéért azt az esetet tekintjük, amikor a két tömeg és a két hossz megegyezik, tehát $m_1 = m_2 = m$ és $l_1 = l_2 = l$. Ekkor a mozgásegyenletek a következő alakúak.

```
> m1 := m2:  m2 := m:  l1 := l2:  l2 := l:
> map( x -> x/(m*l^2), lhs(ode[1]) ):
> ode[1] := map( normal, " ) = 0;
```

$$ode_1 := (3 + 2\cos(\theta_2)) (\frac{\partial^2}{\partial t^2} \theta_1) + (1 + \cos(\theta_2)) (\frac{\partial^2}{\partial t^2} \theta_2)$$
$$- 2 (\frac{\partial}{\partial t} \theta_1) \sin(\theta_2) (\frac{\partial}{\partial t} \theta_2) - (\frac{\partial}{\partial t} \theta_2)^2 \sin(\theta_2) + \frac{g \sin(\theta_1 + \theta_2)}{l}$$
$$+ 2\, \frac{g \sin(\theta_1)}{l} = 0$$

```
> map( x -> x/(m*l^2), lhs(ode[2]) ):
> ode[2] := map( normal, " ) = 0;
```

$$ode_2 :=$$
$$(1 + \cos(\theta_2)) (\frac{\partial^2}{\partial t^2} \theta_1) + (\frac{\partial^2}{\partial t^2} \theta_2) + \sin(\theta_2) (\frac{\partial}{\partial t} \theta_1)^2 + \frac{g \sin(\theta_1 + \theta_2)}{l} = 0$$

Vegyük az $l = 1$ és a $g = 9.8$ értékeket. A kezdeti értékek legyenek

$$\theta_1(0) = 0.04, \quad \theta_2(0) = 0.04, \quad \dot\theta_1(0) = 0.04, \quad \dot\theta_2(0) = 0.04.$$

Most már minden együtt van a kezdetiérték probléma numerikus megoldásához.

```
> l := 1:  g := 9.8:
> alias( t1_0=theta[1](0), t2_0=theta[2](0),
>   t1p_0=D(theta[1])(0), t2p_0=D(theta[2])(0) ):
```

Ismét nem maradt más tennivalónk, mint a `type = numeric` opcióval a **dsolve** eljárás meghívása. Ezúttal a `dverk78` módszert választottuk.

```
> F := dsolve( {ode[1], ode[2], t1_0=0.04, t2_0=0.04,
>   t1p_0=0, t2p_0=0}, {t1,t2}, type = numeric,
>   method=dverk78 );
```

$$F := \mathbf{proc}(dverk78_t) \ldots \mathbf{end}$$

```
> F(0.5); # an example
```

$$[t = .5, \theta_1 = .02561531652085891, \frac{\partial}{\partial t}\theta_1 = -.09576338917809546,$$
$$\theta_2 = -.01205989941048582, \frac{\partial}{\partial t}\theta_2 = -.07345182765453390]$$

A numerikus megoldó algoritmus olyan listát ad vissza, amely öt egyenlőséget tartalmaz: az első a t független változó értékét adja meg, a következő kettő a θ_1 és a θ_2 függő változók értékét ezen a helyen, a két utolsó pedig pedig a függő változók deriváltjait ugyanitt.

```
> Theta[1] := t -> rhs( op(2,F(t)) );
```

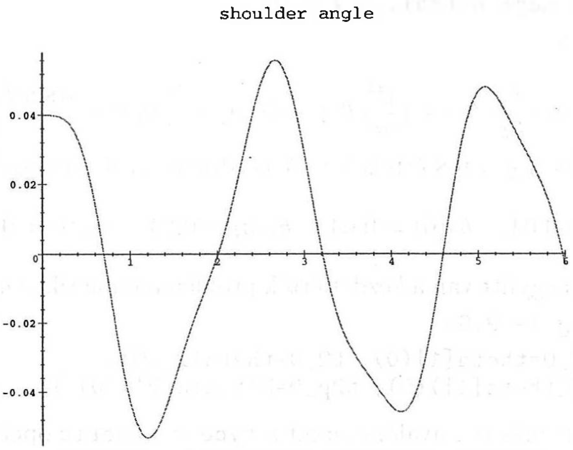
$$\Theta_1 := t \to \mathrm{rhs}(\mathrm{op}(2, \mathrm{F}(t)))$$

```
> Theta[2] := t -> rhs( op(4,F(t)) );
```

$$\Theta_2 := t \to \mathrm{rhs}(\mathrm{op}(4, \mathrm{F}(t)))$$

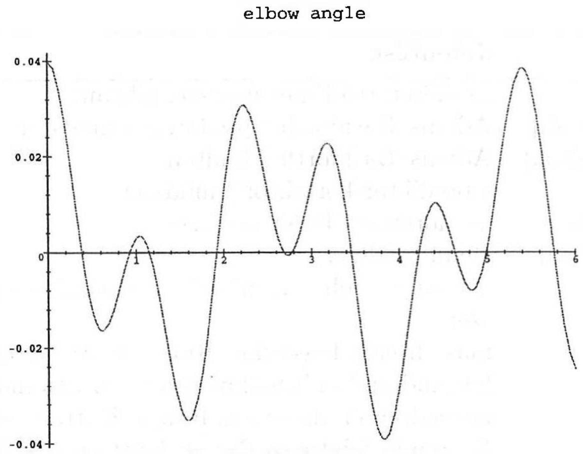
Fölrajzoltatjuk a θ_1-re és a θ_2-re kapott közelítést, hogy legyen valami elképzelésünk a kapott megoldásról (17.7. és 17.8. ábra).

```
> plot( Theta[1], 0..6, title = 'shoulder angle' );
```



17.7. ábra: A kettős inga θ_1 vállszögének grafikonja

```
> plot( Theta[2], 0..6, title = 'elbow angle' );
```



17.8. ábra: A kettős inga θ_2 könyökszögének grafikonja

A szögekre vonatkozóan ilyen eredményt várt az Olvasó? A `plots` csomag **odeplot** eljárásával a (θ_1, θ_2) szögeket ábrázoló háromdimenziós megoldásgörbe is készíthető. A nézőpont megfelelő változtatásával ebből megkaphatjuk a megoldásgörbének az előző két ábrán látható kétdimenziós projekcióit. Animáció segítségével még jobban megérthetjük a kettős inga mozgását. Az alábbiakban megadjuk az animációt generáló parancsokat. Ügyeljünk arra, hogy ezek az előző számítások folytatását képezik.

Először a robotkar „könyökének" és végének pozícióját leíró függvényeket definiáljuk.

```
> x1 := t -> sin( Theta[1](t) ):
> y1 := t -> cos( Theta[1](t) ):
> x2 := t -> sin( Theta[1](t) ) +
>   sin( Theta[1](t) + Theta[2](t) ):
> y2 := t -> cos( Theta[1](t) ) +
>   cos( Theta[1](t) + Theta[2](t) ):
```

Ezután olyan ábrasorozatot generálunk, amely a $(0, 6)$ intervallumból vett különböző időpontokban ábrázolja a robotkar pillanatnyi helyzetét.

```
> for i from 0 to 6 by 1/10 do
>   P[i] := plot( [ [0,0], [x1(i),y1(i)], [x2(i),y2(i)] ],
>   style=line, view=[-0.2..0.2,0..2] )
> od: plotsequence := [ seq( P[i/10], i=0..60 ) ]:
```

Végül a `plots` csomag **display** rutinjával folyamatos sorozatként jelenítjük meg az ábrákat, így kapjuk meg az animációt.

```
> plots[display]( plotsequence, insequence=true );
```

A KDE-ek numerikus megoldását végző két Runge–Kutta módszerrel már találkoztunk. De a Maple ennél sokkal több numerikus módszert tartalmaz. A 17.4. táblázat összegzi a **dsolve** segítségével elérhető módszereket. Ehhez csak a `type = numeric` és a `method =` *approach* opciókat kell megadnunk.

Módszer	Jelentése
classical	az előretartó Euler módszer (default)
classical[adambash]	Adams–Bashforth („prediktor”) módszer
classical[abmoulton]	Adams–Bashforth–Moulton („prediktor-korrektor”) módszer
classical[foreuler]	az előretartó Euler módszer
classical[heunform]	Heun módszer
classical[impoly]	a javított poligon, másnéven módosított Euler módszer
classical[rk2]	másodrendű klasszikus Runge–Kutta módszer
classical[rk3]	harmadrendű klasszikus Runge–Kutta módszer
classical[rk4]	negyedrendű klasszikus Runge–Kutta módszer
dverk78	7,8 rendű folytonos Runge–Kutta módszer
gear	a Gear féle egylépéses extrapolációs módszer
gear[bstoer]	a Bulirsch–Stoer racionális extrapolációs módszer
gear[polyextr]	polinomiális extrapolációs módszer
lsode	Livermore stiff KDE megoldó alg.
lsode[adamsband]	chord iterációt végző implicit Adams módszer sávos Jacobi-mátrixra
lsode[adamsdiag]	chord iterációt végző implicit Adams módszer diagonális Jacobi-mátrixra
lsode[adamsfunc]	implicit Adams módszer függvényiterációval (default)
lsode[adamsfull]	chord iterációt végző implicit Adams módszer teljesen kitöltött Jacobi-mátrixra
lsode[backband]	a hátrafelé differenciáló formulák módszere sávos Jacobi-mátrixra
lsode[backdiag]	a hátrafelé differenciáló formulák módszere diagonális Jacobi-mátrixra
lsode[backfunc]	a hátrafelé differenciáló formulák módszere függvényiterációval
lsode[backfull]	a hátrafelé differenciáló formulák módszere teljesen kitöltött Jacobi-mátrixra
mgear	Gear többlépéses extrapolációs módszere
mgear[adamspc]	Adams féle prediktor-korrektor módszer
mgear[msteppart]	többlépéses módszer stiff rendszerekre a Jacobi-mátrix lépésenkénti kiértékelésével
mgear[mstepnum]	többlépéses módszer stiff rendszerekre a Jacobi-mátrix numerikus differenciálással számított közelítő értékeivel (default)
rkf45	Fehlberg 4,5 rendű Runge–Kutta módszere
taylorseries	a numerikus Taylor-sor módszer

17.4. táblázat: A **dsolve**-val elérhető numerikus módszerek

A klasszikus numerikus módszerek leírása megtalálható majdnem minden, a KDE-k numerikus megoldásával foglalkozó könyvben. A Gear féle egy- és többlépéses módszerekkel kapcsolatban [70]-re hivatkozunk. A GEAR kód további javítása az LSODE (Livermore Stiff ODE Solver), amely az ODEPACK nevű nagyobb programcsomag része (lásd [102]). Az elsőrendű differenciálegyenletek rendszereit Taylor sor módszerrel numerikusan megoldó eljárást a [11] irodalom ismerteti.

A KDE-k numerikus megoldását végző különféle eljárások hívásával kapcsolatos különféle kérdéseinkre a rendszer online help oldalai nyújtanak gyors segítséget. Most csak a Taylor sor módszer alkalmazását mutatjuk be az anharmonikus oszcillátort reprezentáló alábbi differenciálegyenletek rendszerére.

```
> ODEs := diff(q(t),t) = p(t),
>   diff(p(t),t) = -q(t) - 4*q(t)^2 - 2*q(t)^3;
```

$$ODEs := \frac{\partial}{\partial t}\,\mathrm{q}(t) = \mathrm{p}(t),\ \frac{\partial}{\partial t}\,\mathrm{p}(t) = -\mathrm{q}(t) - 4\,\mathrm{q}(t)^2 - 2\,\mathrm{q}(t)^3$$

A kezdeti föltételek legyenek a következők:

```
> initvals := q(0)=0, p(0)=1:
```

```
> dsolve( { ODEs, initvals }, {q(t),p(t)},
>   type=numeric, method=taylorseries,
>   output=listprocedure );
```

$$[\,t = (\mathbf{proc}(t)\ \ldots\ \mathbf{end}),\ \mathrm{q}(t) = (\mathbf{proc}(t)\ \ldots\ \mathbf{end}),\ \mathrm{p}(t) = (\mathbf{proc}(t)\ \ldots\ \mathbf{end})\,]$$

A numerikus megoldó eljárásnak azért adtuk meg az `output = listprocedure` opciót, hogy eredményül olyan egyenlőségek listáját kapjuk, amelyek baloldalán a független és a függő változók nevei állnak, a jobb oldalon pedig az ezeket kiszámító eljárások. Így könnyen kaphatunk numerikus függvényeket a q pozícióra és a p momentumra:

```
> Q := subs(",q(t)); # position
```

$$Q := \mathbf{proc}(t)\ \ldots\ \mathbf{end}$$

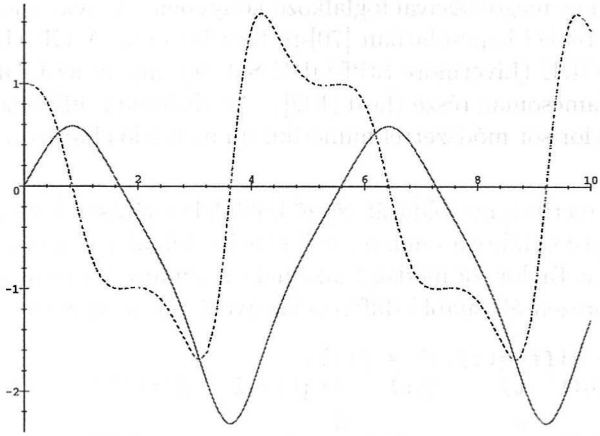
```
> P := subs("",p(t)):  # momentum
```

A pozíció és a momentum időbeli változását láthatjuk a 17.9. ábrán. Ezt a következő parancsokkal számítottuk ki:

```
> Qplot := plot( Q, 0..10, linestyle=1 ):
```

```
> Pplot := plot( P, 0..10, linestyle=2 ):
```

```
> plots[display]( {Qplot, Pplot} );
```



17.9. ábra: Az anharmonikus oszcillátor pozíciójának és momentumának grafikonja

Könnyen kaphatunk numerikus közelítést az anharmonikus oszcillátor periódusára is.

```
> fsolve( Q, 5..6 );
```

$$5.583431954$$

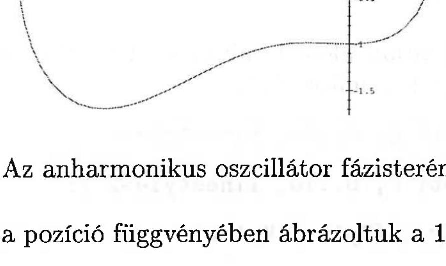
Ellenőrzésül nézzük meg a pozíciót öt rezgés után.

```
> Q(5*");
```

$$.1207517867\,10^{-7}$$

Az oszcillátor fázistérbeli (momentum/pozíció) pályagörbéje a 17.10. ábrán látható.

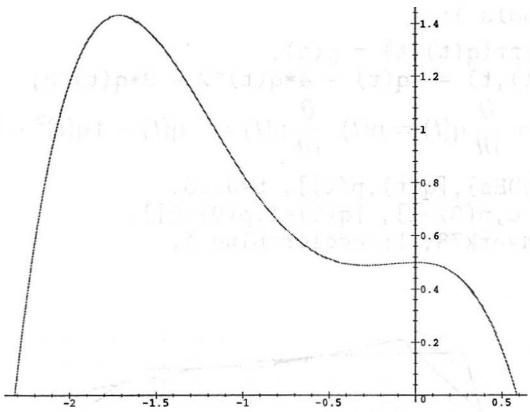
```
> plot( [ Q, P, 0..5.6 ] ); # phase plot
```



17.10. ábra: Az anharmonikus oszcillátor fázisterének ábrázolása

A mozgási energiát a pozíció függvényében ábrázoltuk a 17.11. ábrán.

```
> plot([ Q, P^2/2, 0..5.6 ] );
```



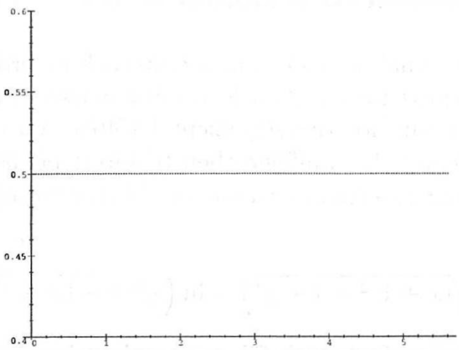
17.11. ábra: Az anharmonikus oszcillátor mozgási energiája a pozíció függvényében

A 17.12. ábra azt mutatja, hogy az oszcillátor

$$H = \frac{1}{2}p^2 + \frac{1}{2}q^2 + \frac{4}{3}q^3 + \frac{1}{2}q^4$$

Hamilton-függvénye konstans.

```
> plot( 1/2*P^2 + 1/2*Q^2 + 4/3*Q^3 + 1/2*Q^4,
>   0..5.6, 0.4..0.6, numpoints=25, adaptive=false );
```



17.12. ábra: A mozgás konstans Hamilton-függvénye

17.6. A DEtools csomag

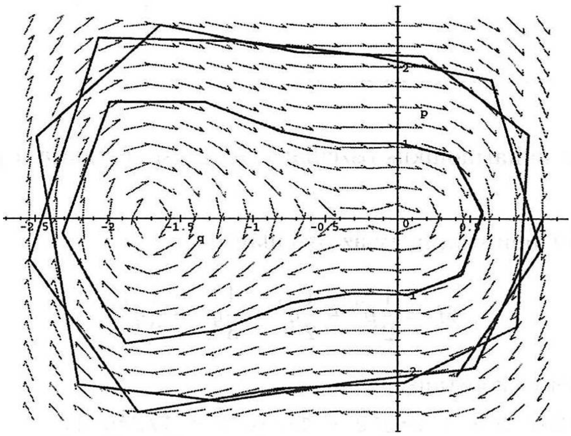
A DEtools csomag a differenciálegyenletek numerikus viselkedésének tanulmányozását segítő grafikus eszközöket tartalmaz. A következő parancs például az

előző részben vizsgált anharmonikus oszcillátor fázisterét jeleníti meg két integrálgörbe berajzolásával (17.13. ábra).

```
> with( DEtools ):
> ODEs := diff(q(t),t) = p(t),
>   diff(p(t),t) = -q(t) - 4*q(t)^2 - 2*q(t)^3;
```

$$ODEs := \frac{\partial}{\partial t}\,\mathrm{q}(t) = \mathrm{p}(t),\ \frac{\partial}{\partial t}\,\mathrm{p}(t) = -\mathrm{q}(t) - 4\,\mathrm{q}(t)^2 - 2\,\mathrm{q}(t)^3$$

```
> DEplot( {ODEs},[q(t),p(t)], t=0..8,
>   [ [q(0)=0,p(0)=1], [q(0)=1,p(0)=0]],
>   method=dverk78, linecolor=blue );
```



17.13. ábra: Az anharmonikus oszcillátor fázistere

Második példaként annak az elektromos erőtérnek az erővonalait számítjuk ki, amelyet két egységnyi távolságban lévő párhuzamos egyeneseken elhelyezkedő ellentétes polaritású, hosszegységenként 1 töltésegység nagyságú töltések hoznak létre. Valójában a 15.5. alfejezetben tekintett példát vettük ismét elő. A töltéseloszlás által az xy-síkban létrehozott elektromos mezőt $\vec{E} = -\nabla\phi$ definiálja, ahol

$$\phi = \ln\left(\sqrt{(x+1)^2+1+y^2}\right) - \ln\left(\sqrt{(x-1)^2+1+y^2}\right).$$

A tér egy erővonalát úgy foghatjuk föl, mint valamely egységnyi töltésnek az elektromos mező hatására történő helyváltoztatásának pályáját. Kiszámítása a $\dfrac{dx}{dt} = E_x$, $\dfrac{dy}{dt} = E_y$ differenciálegyenletek megoldásával történhet, melyekben t a pályagörbe menti elmozdulást jelenti.

```
> phi := ln(sqrt((x+1)^2+y^2)) - ln(sqrt((x-1)^2+y^2));
```

$$\phi := \ln\left(\sqrt{x^2+2\,x+1+y^2}\right) - \ln\left(\sqrt{x^2-2\,x+1+y^2}\right)$$

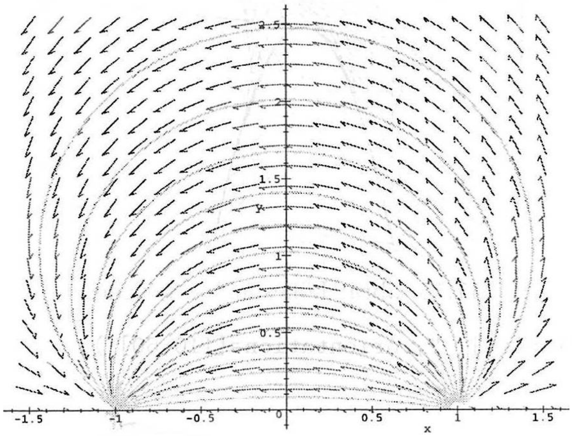
```
> E := map( normal, linalg[grad]( -phi, [x,y] ) );
```

$$E := \left[2\,\frac{x^2-1-y^2}{(x^2+2\,x+1+y^2)\,(x^2-2\,x+1+y^2)},\; 4\,\frac{y\,x}{(x^2+2\,x+1+y^2)\,(x^2-2\,x+1+y^2)}\right]$$

```
> alias( x=x(t), y=y(t) ):
> ODEs := diff(x,t)=E[1], diff(y,t)=E[2];
```

$$ODEs := \frac{\partial}{\partial t}x = 2\,\frac{x^2-1-y^2}{(x^2+2\,x+1+y^2)\,(x^2-2\,x+1+y^2)},\; \frac{\partial}{\partial t}y = 4\,\frac{y\,x}{(x^2+2\,x+1+y^2)\,(x^2-2\,x+1+y^2)}$$

```
> initvals := subs( t=0, [ seq( [x=1+0.05*cos(Pi/20*s),
>   y=.05*sin(Pi/20*s)], s=5..19 ) ] ):
> initvals := evalf( initvals ):
> DEplot( {ODEs}, [x,y], t=0..20, initvals,
>   x=-1.5..1.5, y=0..2.5, stepsize=0.001 );
```



17.14. ábra: Eletromos mező képe

Az ábrázolás tartományát és a lépésközt úgy választottuk, hogy a 17.14. ábrán látható erővonalak elég közel kerüljenek az eletrosztatikus potenciál $(-1, 0)$ szinguláris pontjához. Ennek az a hátránya, hogy egész sok időre és memóriára van szükség az erővonalak kiszámításához.

A **PDEplot** eljárással

$$P(x,t,u)\frac{\partial u}{\partial x} + Q(x,t,u)\frac{\partial u}{\partial t} = R(x,t,u)$$

alakú elsőrendű kvázilineáris parciális differenciálegyenletek numerikus megoldásait ábrázolhatjuk. Példaként tekintsük azokat a folyadékokban kialakuló lökéshullámokat, amelyek a

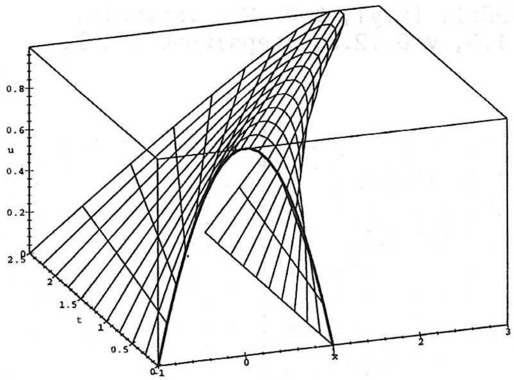
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

PDE és a következő kezdeti értékek hatásának felelnek meg:

$$u(x,0) = \begin{cases} 1 - x^2, & \text{ha } |x| \leq 1; \\ 0 & \text{egyébként.} \end{cases}$$

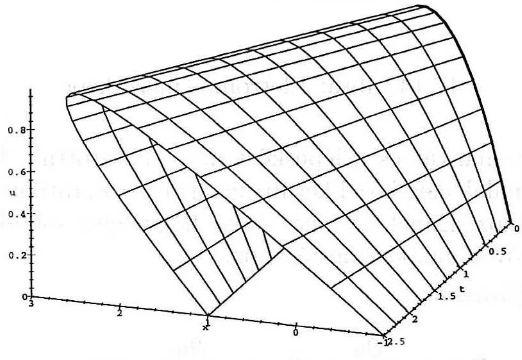
A lökéshullámok terjedése a 17.15. és a 17.16. ábrán látható.

```
> alias( u=u(x,t) ):
> PDE := diff(u,t) + u*diff(u,x) = 0:
> initdata := [s,0,1-s^2]:
> PDEplot( PDE, u, initdata, s=-1..1, x=-1..3, t=0..2.5,
>   orientation=[-110,60], axes=box, style=hidden,
>   shading=none );
```



17.15. ábra: Lökéshullám terjedése

```
> plots[display]( ", orientation=[110,60] );
```



17.16. ábra: Lökéshullám terjedése

A DEtools csomag másik fontos alkalmazása a differenciálegyenletekben szereplő változók helyettesítése. Mind a független, mind a függő változókat transzformálhatjuk. A megfelelő eljárások neve **Dchangevar** és **PDEchangecoords**.

A KDE-ben végrehajtott változótranszformáció példájaként tekintsük a négyzetes potenciáltérben mozgó részecske Schrödinger-egyenletét.

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}+\frac{1}{2}kx^2\psi=E\psi.$$

```
> ODE := -h^2/(2*m)*diff(psi(x),[x$2]) +
>    1/2*k*x^2*psi(x) = E*psi(x);
```

$$ODE := -\frac{1}{2}\frac{h^2\,(\frac{\partial^2}{\partial x^2}\psi(x))}{m}+\frac{1}{2}k\,x^2\,\psi(x)=E\,\psi(x)$$

```
> a := sqrt(h/(m*omega));
```

$$a := \sqrt{\frac{h}{m\,\omega}}$$

```
> E := h/2*omega*lambda;
```

$$E := \frac{1}{2}h\,\omega\,\lambda$$

```
> omega := sqrt(k/m); # oscillator frequency
```

$$\omega := \sqrt{\frac{k}{m}}$$

```
> Dchangevar( { x=a*xi, psi(x)=phi(xi) }, ODE, x, xi );
```

$$-\frac{1}{2}h\sqrt{\frac{k}{m}}\,(\frac{\partial^2}{\partial\xi^2}\phi(\xi))+\frac{1}{2}\frac{k\,h\,\xi^2\,\phi(\xi)}{m\sqrt{\frac{k}{m}}}=\frac{1}{2}h\sqrt{\frac{k}{m}}\,\lambda\,\phi(\xi)$$

```
> simplify( (rhs-lhs)("), assume=positive );
```

$$-\frac{1}{2}\frac{h\sqrt{k}\,(-\lambda\,\phi(\xi)-(\frac{\partial^2}{\partial\xi^2}\phi(\xi))+\xi^2\,\phi(\xi))}{\sqrt{m}}$$

```
> select( has, ", phi );
```

$$-\lambda\,\phi(\xi)-(\frac{\partial^2}{\partial\xi^2}\phi(\xi))+\xi^2\,\phi(\xi)$$

```
> newODE := collect( ", phi );
```

$$newODE := (-\lambda+\xi^2)\,\phi(\xi)-(\frac{\partial^2}{\partial\xi^2}\phi(\xi))$$

```
> Dchangevar( phi(xi) = exp(-1/2*xi^2)*y(xi), newODE, xi );
```

$$(-\lambda+\xi^2)\,e^{(-1/2\,\xi^2)}\,\mathrm{y}(\xi)+e^{(-1/2\,\xi^2)}\,\mathrm{y}(\xi)-\xi^2\,e^{(-1/2\,\xi^2)}\,\mathrm{y}(\xi)$$
$$+2\,\xi\,e^{(-1/2\,\xi^2)}\,(\frac{\partial}{\partial\xi}\mathrm{y}(\xi))-e^{(-1/2\,\xi^2)}\,(\frac{\partial^2}{\partial\xi^2}\mathrm{y}(\xi))$$

```
> factor(");
```

$$e^{(-1/2\,\xi^2)}\,(-\lambda\,\mathrm{y}(\xi)+\mathrm{y}(\xi)+2\,\xi\,(\frac{\partial}{\partial\xi}\,\mathrm{y}(\xi))-(\frac{\partial^2}{\partial\xi^2}\,\mathrm{y}(\xi)))$$

```
> select( has, ", y );
```

$$-\lambda\,\mathrm{y}(\xi)+\mathrm{y}(\xi)+2\,\xi\,(\frac{\partial}{\partial\xi}\,\mathrm{y}(\xi))-(\frac{\partial^2}{\partial\xi^2}\,\mathrm{y}(\xi))$$

```
> HermiteODE := collect( ", y );
```

$$\mathit{HermiteODE} := (-\lambda+1)\,\mathrm{y}(\xi)+2\,\xi\,(\frac{\partial}{\partial\xi}\,\mathrm{y}(\xi))-(\frac{\partial^2}{\partial\xi^2}\,\mathrm{y}(\xi))$$

A differenciálegyenlet hatványsorral fölírt megoldását már a 17.4. alfejezetben kiszámítottuk.

A **PDEchangecoords** eljárással úgy változtathatjuk meg a PDE-ben szereplő koordinátákat, mint ahogy a plot rutinokban váltottunk koordinátarendszert. A kétdimenziós Laplace egyenletet például a következő módon írhatjuk át polárkoordinátás alakra:

```
> LaplacePDE := diff(F(x,y),[x$2])
> + diff(F(x,y),[y$2]) = 0;
```

$$\mathit{LaplacePDE} := (\frac{\partial^2}{\partial x^2}\,\mathrm{F}(x,\,y))+(\frac{\partial^2}{\partial y^2}\,\mathrm{F}(x,\,y))=0$$

```
> PDEchangecoords( LaplacePDE, [x,y], polar, [r,phi] );
```

$$\frac{(\frac{\partial}{\partial r}\,\mathrm{F}(r,\,\phi))\,r+(\frac{\partial^2}{\partial r^2}\,\mathrm{F}(r,\,\phi))\,r^2+(\frac{\partial^2}{\partial \phi^2}\,\mathrm{F}(r,\,\phi))}{r^2}=0$$

```
> newLaplacePDE := numer(lhs(")) = 0;
```

$$\mathit{newLaplacePDE} :=$$

$$(\frac{\partial}{\partial r}\,\mathrm{F}(r,\,\phi))\,r+(\frac{\partial^2}{\partial r^2}\,\mathrm{F}(r,\,\phi))\,r^2+(\frac{\partial^2}{\partial \phi^2}\,\mathrm{F}(r,\,\phi))=0$$

Az **addcoords** eljárással mi is bevezethetünk „saját” koordináta-transzformációkat. Ennek illusztrálására transzformáljuk át a

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}+x\frac{\partial u}{\partial x}+u$$

Fokker–Planck egyenletet a

$$\frac{\partial v}{\partial \tau}=\frac{\partial^2 v}{\partial \xi^2}$$

diffúziós egyenletbe.

```
> FockerPlanckPDE := diff(u(x,t),t) =
>   diff(u(x,t),[x$2]) + x*diff(u(x,t),x) + u(x,t);
```

$$\mathit{FockerPlanckPDE} :=$$

$$\frac{\partial}{\partial t}\,\mathrm{u}(x,\,t)=(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x,\,t))+x\,(\frac{\partial}{\partial x}\,\mathrm{u}(x,\,t))+\mathrm{u}(x,\,t)$$

Az első változóhelyettesítések: $\xi = xe^t$, $v = ue^{-t}$.

```
> readlib( addcoords ):
> Dchangevar( {u(x,t)=v(xi,t)*exp(t)},
>   FockerPlanckPDE, x, xi );
```

$$\frac{\partial}{\partial t}\mathrm{v}(\xi, t)\, e^t = (\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, t)\, e^t) + \xi\, (\frac{\partial}{\partial \xi}\mathrm{v}(\xi, t)\, e^t) + \mathrm{v}(\xi, t)\, e^t$$

```
> simplify(");
```

$$(\frac{\partial}{\partial t}\mathrm{v}(\xi, t))\, e^t + \mathrm{v}(\xi, t)\, e^t =$$
$$(\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, t))\, e^t + \xi\, (\frac{\partial}{\partial \xi}\mathrm{v}(\xi, t))\, e^t + \mathrm{v}(\xi, t)\, e^t$$

```
> addcoords( T1, [x,t], [x*exp(-t),t] ):
> PDEchangecoords( "" , [xi,t], T1 );
```

$$(\xi\, e^t - \xi\, e^{(-t)}\, (e^t)^2)\, (\frac{\partial}{\partial \xi}\mathrm{v}(\xi, t)) + (\frac{\partial}{\partial t}\mathrm{v}(\xi, t))\, e^t - (e^t)^3\, (\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, t)) = 0$$

```
> expand( "/exp(t)^3 );
```

$$\frac{\frac{\partial}{\partial t}\mathrm{v}(\xi, t)}{(e^t)^2} - (\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, t)) = 0$$

```
> combine( ", exp );
```

$$e^{(-2\,t)}\, (\frac{\partial}{\partial t}\mathrm{v}(\xi, t)) - (\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, t)) = 0$$

Tehát a következő PDE-t kaptuk:

$$e^{-2t}\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial \xi^2}.$$

Vezessük be a $\tau = \frac{1}{2}e^{2t}$ új változót. Az új koordinátákkal fölírt PDE az egyszerűsítések elvégzése után a „diffúziós egyenlet" néven ismert formájú lesz.

```
> addcoords( T2, [x,t], [x,1/2*ln(2*t)] ):
> PDEchangecoords( "" , [xi,t], T2, [xi,tau] ):
> diffusionPDE := simplify(");
```

$$\mathit{diffusionPDE} := (\frac{\partial}{\partial \tau}\mathrm{v}(\xi, \tau)) - (\frac{\partial^2}{\partial \xi^2}\mathrm{v}(\xi, \tau)) = 0$$

17.7. Perturbációs módszerek

Ha perturbációs módszerekkel szeretnénk közelítő megoldást találni a KDE-hez, a számítógépes algebrai rendszerek a leghasznosabb segédeszközök. Ebben a részben két klasszikus eljárást ismertetünk, a Poincaré–Lindstedt és a többszörös skálázásos módszert. Mindkettőt a van der Pol egyenletre fogjuk alkalmazni. Az érdeklődő Olvasónak ajánljuk a [158,159] irodalmakat is, melyek a MACSYMA számítógépes algebrai rendszernek a perturbáció és a bifurkáció elméletében

való fölhasználásával kapcsolatos számos példát tartalmaznak. A matematikai elmélet megtalálható [144]-ben.

Poincaré–Lindstedt

A van der Pol egyenlet a következő:

$$y'' - \epsilon(1 - y^2)y' + y = 0.$$

Az $\epsilon = 0$ értékre ez a matematikai inga differenciálegyenlete. A 17.5. alfejezetben már numerikusan vizsgáltuk az $\epsilon = 1$ esetet. Az egyenletnek tetszőleges ϵ mellett létezik egy *határciklusnak* nevezett aszimptotikusan stabil periodikus megoldása.

A határciklusra szeretnénk jó numerikus közelítést kapni kis ϵ érték esetén. Mivel a van der Pol egyenlet nem tartalmaz az időtől explicit módon függő tagokat, az általánosság megszorítása nélkül választhatjuk a $t = 0$-nak megfelelő pontot; a kezdeti érték legyen $y(0) = 0$. A Poincaré–Lindstedt módszer alkalmazásakor „megnyújtjuk" az időt a

$$\tau = \omega t$$

transzformációval, amelyben

$$\omega = 1 + \omega_1\epsilon + \omega_2\epsilon^2 + \omega_3\epsilon^3 + \ldots$$

Ekkor az $y(\tau)$-ra fölírt van der Pol egyenlet

$$\omega^2 y'' - \omega\epsilon(1 - y^2)y' + y = 0.$$

Ellenőrizzük mindezt a Maple segítségével.

```
> diff( y(t),t$2 ) - epsilon*(1-y(t)^2)*diff(y(t),t)
>    + y(t) = 0;
```

$$\left(\frac{\partial^2}{\partial t^2}\,\mathrm{y}(t)\right) - \varepsilon\,(1 - \mathrm{y}(t)^2)\left(\frac{\partial}{\partial t}\,\mathrm{y}(t)\right) + \mathrm{y}(t) = 0$$

```
> ODE := DEtools[Dchangevar](
>    { t=tau/omega, y(t)=y(tau) }, ", t, tau );
```

$$ODE := \omega^2\left(\frac{\partial^2}{\partial \tau^2}\,\mathrm{y}(\tau)\right) - \varepsilon\,(1 - \mathrm{y}(\tau)^2)\,\omega\left(\frac{\partial}{\partial \tau}\,\mathrm{y}(\tau)\right) + \mathrm{y}(\tau) = 0$$

Föltesszük, hogy az $y(\tau)$ megoldást ϵ szerint Taylor sorba fejthető, azaz

$$y(\tau) = y_0(\tau) + y_1(\tau)\epsilon + y_2(\tau)\epsilon^2 + y_3(\tau)\epsilon^3 + \ldots$$

Helyettesítsük $y(\tau)$ és $\omega(\epsilon)$ sorfejtését a van der Pol egyenletbe, gyűjtsük össze az ϵ-os tagokat és tegyük nullával egyenlővé ϵ minden hatványának együtthatóját. Kis ϵ esetén a megfelelő egyenletek:

$$\begin{aligned} y_0'' + y_0 &= 0, \\ y_1'' + y_1 &= y_0'(1 - y_0^2) - 2\omega_1 y_0'' \\ y_2'' + y_2 &= (1 - y_0^2)y_1' - 2y_0 y_1 y0' - 2\omega_1 y_1'' \\ &\quad -(2\omega_2 + \omega_1^2)y_0'' + \omega_1(1 - y_0)x_0'. \end{aligned}$$

Az $y(0) = 0$ kezdeti értékből adódnak a következő egyenletek.

$$y_0(0) = 0, \quad y_1(0) = 0 \quad y_2(0) = 0, \quad y_3(0) = 0, \ \ldots$$

Vizsgáljunk meg néhány differenciálegyenletetet a Maple-lel, egyszersmind rögzítsük a szekció további részében alkalmazott jelöléseket is:

```
> e_order := 6:
> macro( e=epsilon, t=tau ):
> alias( seq( y[i] = eta[i](tau), i=0..e_order ) ):
> e := () -> e:  # introduce e as a constant function
> for i from 0 to e_order do
>   eta[i] := t -> eta[i](t)
> od:
> omega := 1 + sum( 'w[i]*e^i', 'i'=1..e_order );
```

$$\omega := 1 + w_1\,\varepsilon + w_2\,\varepsilon^2 + w_3\,\varepsilon^3 + w_4\,\varepsilon^4 + w_5\,\varepsilon^5 + w_6\,\varepsilon^6$$

```
> y := sum( 'eta[i]*e^i', 'i'=0..e_order );
```

$$y := \eta_0 + \eta_1\,\varepsilon + \eta_2\,\varepsilon^2 + \eta_3\,\varepsilon^3 + \eta_4\,\varepsilon^4 + \eta_5\,\varepsilon^5 + \eta_6\,\varepsilon^6$$

```
> deqn := simplify( collect(ODE,e), {e^(e_order+1)=0} ):
> for i from 0 to e_order do
>   ode[i] :=coeff( lhs(deqn), e, i ) = 0
> od:
> ode[0];
```

$$(\frac{\partial^2}{\partial\tau^2}\,y_0) + y_0 = 0$$

```
> ode[1];
```

$$y_1 + (\frac{\partial^2}{\partial\tau^2}\,y_1) + 2\,w_1\,(\frac{\partial^2}{\partial\tau^2}\,y_0) + (\frac{\partial}{\partial\tau}\,y_0)\,{y_0}^2 - (\frac{\partial}{\partial\tau}\,y_0) = 0$$

```
> ode[2];
```

$$y_2 + (\frac{\partial^2}{\partial\tau^2}\,y_2) - (\frac{\partial}{\partial\tau}\,y_0)\,w_1 + (\frac{\partial^2}{\partial\tau^2}\,y_0)\,{w_1}^2 + 2\,(\frac{\partial^2}{\partial\tau^2}\,y_0)\,w_2 + (\frac{\partial}{\partial\tau}\,y_1)\,{y_0}^2$$
$$+ 2\,w_1\,(\frac{\partial^2}{\partial\tau^2}\,y_1) - (\frac{\partial}{\partial\tau}\,y_1) + (\frac{\partial}{\partial\tau}\,y_0)\,w_1\,{y_0}^2 + 2\,(\frac{\partial}{\partial\tau}\,y_0)\,y_0\,y_1 = 0$$

Az $\eta_0(\tau)$-ra vonatkozó kezdetiérték probléma egyszerűen megoldható.

```
> dsolve( { ode[0], eta[0](0)=0, D(eta[0])(0)=C[1] },
>   eta[0](t) );
```

$$y_0 = C_1 \sin(\tau)$$

Ezzel a kifejezéssel definiáljuk az η_0 függvényt, majd az $\eta_1(\tau)$-ra vonatkozó differenciálegyenlet vizsgálatára térünk át.

```
> eta[0] := unapply( rhs("), t );
```

$$\eta_0 := \tau \to C_1 \sin(\tau)$$

```
> ode[1];
```

$$y_1 + (\frac{\partial^2}{\partial\tau^2}\,y_1) - 2\,w_1\,C_1 \sin(\tau) + {C_1}^3 \cos(\tau) \sin(\tau)^2 - C_1 \cos(\tau) = 0$$

```
> map( combine, ode[1], 'trig' );
```

$$y_1 + (\frac{\partial^2}{\partial\tau^2} y_1) - 2\,w_1\,C_1 \sin(\tau) + \frac{1}{4} C_1{}^3 \cos(\tau) - \frac{1}{4} C_1{}^3 \cos(3\,\tau) - C_1 \cos(\tau) = 0$$

```
> ode[1] := map( collect, ", [sin(t),cos(t)] );
```

$ode_1 :=$

$$-2\,w_1\,C_1 \sin(\tau) + (\frac{1}{4} C_1{}^3 - C_1) \cos(\tau) + y_1 + (\frac{\partial^2}{\partial\tau^2} y_1) - \frac{1}{4} C_1{}^3 \cos(3\,\tau) = 0$$

A $\sin\tau$-t és a $\cos\tau$-t tartalmazó tagokat *rezonencia tagoknak* vagy *szekuláris tagoknak* szokás nevezni. Amint az alábbi általános megoldásból is látható, ezek okozzák a közelítés nemperiódikus viselkedését.

```
> dsolve( { ode[1], eta[1](0)=0, D(eta[1])(0)=C[2] },
>   eta[1](t), laplace );
```

$$y_1 = -\frac{1}{32} C_1{}^3 \cos(3\,\tau) + \frac{1}{32} C_1{}^3 \cos(\tau) + C_2 \sin(\tau) + w_1\,C_1 \sin(\tau) - \frac{1}{8} \tau\,C_1{}^3 \sin(\tau) + \frac{1}{2} C_1\,\tau \sin(\tau) - C_1\,\tau\,w_1 \cos(\tau)$$

```
> map( collect, ", [sin(t),cos(t),t] );
```

$$y_1 = ((-\frac{1}{8} C_1{}^3 + \frac{1}{2} C_1)\,\tau + w_1\,C_1 + C_2) \sin(\tau) + (\frac{1}{32} C_1{}^3 - C_1\,\tau\,w_1) \cos(\tau) - \frac{1}{32} C_1{}^3 \cos(3\,\tau)$$

Válasszuk úgy C_1-et és w_1-et, hogy ezek a tagok eltűnjenek.

```
> solve( { coeff( lhs(ode[1]), sin(t) ) = 0,
>   coeff( lhs(ode[1]), cos(t) ) = 0 } );
```

$$\{w_1 = 0, C_1 = 2\}, \{w_1 = 0, C_1 = -2\}, \{C_1 = 0, w_1 = w_1\}$$

Mivel a 17.5. alfejezetben tárgyalt numerikus mószerrel akarjuk összehasonlítani az eredményt, válasszunk negatív amplitudót.

```
> w[1] := 0:  C[1] := -2:
> ode[1];
```

$$y_1 + (\frac{\partial^2}{\partial\tau^2} y_1) + 2 \cos(3\,\tau) = 0$$

Az $\eta_1(0) = 0$, $\eta_1'(0) = C_2$ kezdeti értékekkel megoldjuk az $\eta_1(\tau)$-ra vonatkozó differenciálegyenletet.

```
> dsolve( { ode[1], eta[1](0)=0, D(eta[1])(0)=C[2] },
> eta[1](t), laplace );
```

$$y_1 = \frac{1}{4} \cos(3\,\tau) - \frac{1}{4} \cos(\tau) + C_2 \sin(\tau)$$

```
> eta[1] := unapply( rhs("), tau );
```

$$\eta_1 := \tau \to \frac{1}{4} \cos(3\,\tau) - \frac{1}{4} \cos(\tau) + C_2 \sin(\tau)$$

Hasonlóan járunk el C_2-vel és $\eta_2(\tau)$-val kapcsolatban is.

```
> map( combine, ode[2], 'trig' ):
> ode[2] := map( collect, ",
>   [ sin(t), sin(3*t), cos(t), cos(3*t) ] );
```

$$ode_2 := (\frac{1}{4} + 4\,w_2)\sin(\tau) + y_2 + (\frac{\partial^2}{\partial\tau^2}\,y_2) + 2\,C_2\cos(\tau) - \frac{3}{2}\sin(3\,\tau) + \frac{5}{4}\sin(5\,\tau) - 3\,C_2\cos(3\,\tau) = 0$$

```
> solve( { coeff( lhs(ode[2]), sin(t) ) = 0,
>   coeff( lhs(ode[2]), cos(t) ) = 0 } );
```

$$\{C_2 = 0, w_2 = \frac{-1}{16}\}$$

```
> assign("):
> dsolve( { ode[2], eta[2](0)=0, D(eta[2])(0)=C[3] },
>   eta[2](t), laplace ):
> collect( ", [ sin(t), sin(3*t), sin(5*t),
>   cos(t), cos(3*t), cos(5*t) ] ):
> eta[2] := unapply( rhs("), t );
```

$$\eta_2 := \tau \to (C_3 + \frac{29}{96})\sin(\tau) + \frac{5}{96}\sin(5\,\tau) - \frac{3}{16}\sin(3\,\tau)$$

Föltételezzük, hogy az Olvasó megérti a magasabb rendű tagok kiszámításában mutatkozó szabályosságot, és az ezt kihasználó alábbi ciklikus utasításokat.

```
> for i from 3 to e_order do
>   map( combine, ode[i], 'trig' ):
>   ode[i] := map( collect, ",
>     [ seq(sin((2*j+1)*t),j=0..i),
>     seq(cos((2*j+1)*t),j=0..i) ] ):
>   solve( {coeff( lhs(ode[i]), sin(t) ) = 0,
>   coeff( lhs(ode[i]), cos(t) ) = 0} ):
>   assign("):
>   dsolve( { ode[i], eta[i](0)=0, D(eta[i])(0)=C[i+1] },
>     eta[i](t), laplace ):
>   collect( ", [ seq(sin((2*j+1)*t),j=0..i),
>     seq(cos((2*j+1)*t),j=0..i) ] ):
>   eta[i] := unapply( rhs("), t )
> od:
```

Vessünk egy pillantást a végeredményre.

```
> omega;
```

$$1 - \frac{1}{16}\,\varepsilon^2 + \frac{17}{3072}\,\varepsilon^4 + \frac{35}{884736}\,\varepsilon^6$$

```
> y(t);
```

$$
\begin{aligned}
&-2\sin(\tau)+(\frac{1}{4}\cos(3\,\tau)-\frac{1}{4}\cos(\tau))\,\varepsilon+(\frac{5}{96}\sin(5\,\tau)-\frac{3}{16}\sin(3\,\tau))\,\varepsilon^2\\
&+(-\frac{21}{256}\cos(3\,\tau)-\frac{7}{576}\cos(7\,\tau)+\frac{5}{72}\cos(5\,\tau)+\frac{19}{768}\cos(\tau))\,\varepsilon^3+(\\
&-\frac{11}{4096}\sin(\tau)+\frac{29}{768}\sin(3\,\tau)+\frac{2555}{110592}\sin(7\,\tau)-\frac{61}{20480}\sin(9\,\tau)\\
&-\frac{1385}{27648}\sin(5\,\tau))\varepsilon^4+(\frac{5533}{7372800}\cos(11\,\tau)+\frac{4175}{294912}\cos(3\,\tau)\\
&+\frac{153251}{6635520}\cos(7\,\tau)-\frac{9013}{1228800}\cos(9\,\tau)-\frac{77915}{2654208}\cos(5\,\tau)\\
&-\frac{5807}{4423680}\cos(\tau))\varepsilon^5+((C_7-\frac{148447039}{55738368000})\sin(\tau)\\
&+\frac{715247}{3715891200}\sin(13\,\tau)-\frac{6871193}{398131200}\sin(7\,\tau)+\frac{690583}{73728000}\sin(9\,\tau)\\
&-\frac{143191}{35389440}\sin(3\,\tau)+\frac{469795}{31850496}\sin(5\,\tau)-\frac{6017803}{2654208000}\sin(11\,\tau))\varepsilon^6
\end{aligned}
$$

A C_7 konstanst a következő lépésben tudnánk meghatározni. Az $y(\tau)$ megoldást csak az ötödrendű tagokkal bezárólag írjuk föl; a [6] irodalomban egész 164 rendig megtalálható a számítás eredménye.

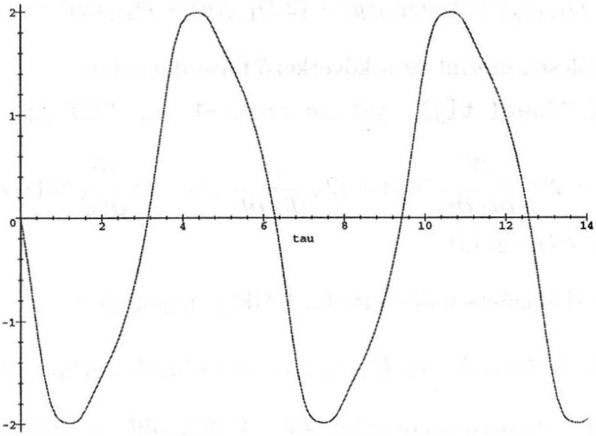
```
> y := unapply( simplify( y(t), {e^e_order=0} ), t ):
```

Ábrázoljuk ezt a függvényt az $\epsilon = 1$ esetben.

```
> e := 1:  y(t);
```

$$
\begin{aligned}
&-\frac{799991}{1105920}\cos(\tau)+\frac{5257957}{3317760}\cos(\tau)^3-\frac{81}{32}\sin(\tau)\cos(\tau)^4\\
&+\frac{13}{256}\sin(\tau)\cos(\tau)^2-\frac{1037927}{552960}\sin(\tau)+\frac{912187}{129600}\cos(\tau)^7\\
&-\frac{1212373}{259200}\cos(\tau)^5-\frac{61}{80}\sin(\tau)\cos(\tau)^8+\frac{1519}{540}\sin(\tau)\cos(\tau)^6\\
&-\frac{114941}{28800}\cos(\tau)^9+\frac{5533}{7200}\cos(\tau)^{11}
\end{aligned}
$$

```
> plot( y(t), t=0..14 );
```



17.17. ábra: A van der Pol egyenlet megoldása a Poincaré–Lindstedt módszerrel

Hasonlítsuk össze mindezt a 17.5. alfejezet numerikus megoldásával.

A többszörös skálázásos módszer

A Poincaré–Lindstedt módszer jól alkalmazható KDE-k periodikus megoldásainak meghatározására, de nem sokat segít a KDE határciklusának közelébe eső általános megoldások közelítésénél. A többszörös skálázásos módszer használatakor a KDE-t nem egyetlen t változó függvényének tekintjük. Ehelyett kettő vagy több $t_1, t_2, t_3, \ldots$ független változót veszünk, amelyeket a

$$t_1 = t, \quad t_2 = \epsilon t, \quad t_3 = \epsilon^2 t, \ \ldots$$

egyenlőségekkel definiálunk. Ekkor

$$\begin{aligned}\frac{d}{dt} &= \frac{\partial}{\partial t_1} + \epsilon\frac{\partial}{\partial t_2} + \epsilon^2\frac{\partial}{\partial t_3} + \cdots \\ \frac{d^2}{dt^2} &= \frac{\partial^2}{\partial t_1^2} + 2\epsilon\frac{\partial^2}{\partial t_1 t_2} + \epsilon^2\left(\frac{\partial^2}{\partial t_2^2} + 2\frac{\partial}{\partial t_3}\right) + \cdots .\end{aligned}$$

Ezt a Maple segítségével is ellenőrizhetjük (a magasabb rendű tagokat az e_order változó értékének megnövelésével kaphatjuk meg).

```
> restart;
> alias( epsilon=e,
>   seq( y[i] = eta[i]( seq(t[j],j=1..3) ), i=0..2 ) ):
> macro( t1=t[1], t2=t[2], t3=t[3] ):
> e_order := 2:
> e := subs( variables = seq( u.j, j=0..e_order),
>   body=e, (variables -> body) ):
> subs( D = sum( e^(i-1)*D[i], i=1..e_order+1 ),
>   (D@@e_order)(y) ):
```

```
> simplify( collect(",e), {e^(e_order+1)=0} );
```

$$D_{1,1}(y)+2\varepsilon\,D_{1,2}(y)+(2\,D_{1,3}(y)+D_{2,2}(y))\,\varepsilon^2$$

A klasszikus jelölések szerint ez a következő formulát adja:

```
> convert( "(seq( t[j], j=1..e_order+1 )), diff );
```

$$\Big(\frac{\partial^2}{\partial t_1^{\,2}}\,\%1\Big)+2\varepsilon\Big(\frac{\partial^2}{\partial t_1\,\partial t_2}\,\%1\Big)+\Big(2\Big(\frac{\partial^2}{\partial t_1\,\partial t_3}\,\%1\Big)+\Big(\frac{\partial^2}{\partial t_2^{\,2}}\,\%1\Big)\Big)\varepsilon^2$$
$$\%1 := \mathrm{y}(t_1,\,t_2,\,t_3)$$

A többszörös skálázásos módszernél a KDE y megoldását

$$y=y_0(t_1,t_2,t_3,\ldots)+\epsilon y_1(t_1,t_2,t_3,\ldots)+\epsilon^2 y_2(t_1,t_2,t_3,\ldots)+\cdots$$

alakú ϵ szerinti hatványsorként írjuk föl. Ebből differenciálegyenleteket kapunk az y_0, y_1, y_2, stb. függvényekre. A három változós esetet alkalmazzuk az $y(0)=0$, $y'(0)=-1/10$ kezdeti értékekkel az

$$y''-\epsilon(1-y^2)y'+y=0$$

van der Pol egyenletre. Valójában Noble és Hussain [145]-ben leírt számításait ismételjük meg.

Először az $y_0(t_1,t_2,t_3)$, $y_1(t_1,t_2,t_3)$, és $y_2(t_1,t_2,t_3)$ függvényekre vonatkozó differenciálegyenleteket vezetjük le a Maple-lel.

```
> ODE := (D@@2)(y) - e*(1-y^2)*D(y) + y=0;
```

$$ODE := (D^{(2)})(y)-\varepsilon\,(1-y^2)\,\mathrm{D}(y)+y=0$$

```
> subs( D = sum('e^(i-1)*D[i]','i'=1..e_order+1), ODE ):
> y := sum( 'eta[i]*e^i', 'i'=0..e_order );
```

$$y := \eta_0+\eta_1\,\varepsilon+\eta_2\,\varepsilon^2$$

```
> diffeqn := simplify( collect("",e), {e^(e_order+1)=0} ):
> for i from 0 to e_order do
> ode[i] := coeff( lhs(diffeqn), e, i ) = 0
> od;
```

$$ode_0 := D_{1,1}(\eta_0)+\eta_0=0$$

$$ode_1 := D_{1,1}(\eta_1)+2\,D_{1,2}(\eta_0)+\eta_1-D_1(\eta_0)+D_1(\eta_0)\,{\eta_0}^2=0$$

$$\begin{aligned}ode_2 := {}&-D_1(\eta_1)-D_2(\eta_0)+{\eta_0}^2\,D_1(\eta_1)+{\eta_0}^2\,D_2(\eta_0)\\&+2\,\eta_0\,\eta_1\,D_1(\eta_0)+2\,D_{1,2}(\eta_1)+\eta_2+D_{2,2}(\eta_0)+D_{1,1}(\eta_2)\\&+2\,D_{1,3}(\eta_0)=0\end{aligned}$$

A keresett differenciálegyenletek tehát a következők:

$$\begin{aligned}
\frac{\partial^2 y_0}{\partial t_1^2} + y_0 &= 0,\\
\frac{\partial^2 y_1}{\partial t_1^2} + y_1 &= -2\frac{\partial^2 y_0}{\partial t_1 t_2} + (1-y_0^2)\frac{\partial y_0}{\partial t_1},\\
\frac{\partial^2 y_2}{\partial t_1^2} + y_2 &= -2\frac{\partial^2 y_0}{\partial t_1 t_2} - \frac{\partial^2 y_0}{\partial t_2^2} - 2\frac{\partial^2 y_0}{\partial t_1 t_3} +\\
&\quad (1-y_0^2)\left(\frac{\partial y_1}{\partial t_1} + \frac{\partial y_0}{\partial t_2}\right) - 2y_0 y_1 \frac{\partial y_0}{\partial t_1}.
\end{aligned}$$

Az első egyenlet általános megoldása

$$y_0(t_1, t_2, t_3) = A(t_2, t_3)\sin(t_1 + B(t_2, t_3)).$$

Ezt a következő egyenletbe helyettesítve a rezonancia tagok elhagyása után $A(t_2, t_3)$-ra és $B(t_2, t_3)$-ra a következő differenciálegyenleteket kapjuk:

$$\begin{aligned}
\frac{\partial B(t_2, t_3)}{\partial t_2} &= 0,\\
\frac{\partial A(t_2, t_3)}{\partial t_2} &= \frac{1}{2}A(t_2, t_3) - \frac{1}{8}A(t_2, t_3)^3.
\end{aligned}$$

Ellenőrizzük a levezetéseket a Maple segítségével.

```
> ode[1] := convert( ode[1](t1,t2,t3), diff );
```

$$ode_1 := \left(\frac{\partial^2}{\partial t_1^2} y_1\right) + 2\left(\frac{\partial^2}{\partial t_1\,\partial t_2} y_0\right) + y_1 - \left(\frac{\partial}{\partial t_1} y_0\right) + \left(\frac{\partial}{\partial t_1} y_0\right) y_0^2 = 0$$

```
> eta[0] := (t1,t2,t3) -> A(t2,t3) * sin(t1 + B(t2,t3));
```

$$\eta_0 := (t1, t2, t3) \to \mathrm{A}(t2, t3)\sin(t1 + \mathrm{B}(t2, t3))$$

```
> combine( ode[1], 'trig' ):
> ode[1] := collect( ", [ sin(t1+B(t2,t3)),
>   cos(t1+B(t2,t3)) ] );
```

$$\begin{aligned}
ode_1 := &-2\,\mathrm{A}(t_2, t_3)\sin(t_1 + \mathrm{B}(t_2, t_3))\left(\frac{\partial}{\partial t_2}\mathrm{B}(t_2, t_3)\right)\\
&+ \left(2\left(\frac{\partial}{\partial t_2}\mathrm{A}(t_2, t_3)\right) - \mathrm{A}(t_2, t_3) + \frac{1}{4}\mathrm{A}(t_2, t_3)^3\right)\cos(t_1 + \mathrm{B}(t_2, t_3))\\
&+ \left(\frac{\partial^2}{\partial t_1^2} y_1\right) - \frac{1}{4}\mathrm{A}(t_2, t_3)^3\cos(3\,t_1 + 3\,\mathrm{B}(t_2, t_3)) + y_1 = 0
\end{aligned}$$

A rezonancia tagok

$$\sin(t_1 + B(t_2, t_3)), \cos(t_1 + B(t_2, t_3)).$$

Ha továbbra is megkívánjuk, hogy a rezonancia tagok együtthatója nulla legyen, a következő differenciálegyenleteket kapjuk.

```
> restrictions := {
>   coeff( lhs(ode[1]), cos(t1+B(t2,t3)) ) = 0,
>   coeff( lhs(ode[1]), sin(t1+B(t2,t3)) ) = 0 };
```

$$restrictions := \left\{ -2\,\mathrm{A}(t_2,\,t_3)\,(\frac{\partial}{\partial t_2}\,\mathrm{B}(t_2,\,t_3)) = 0,\right.$$
$$\left. 2\,(\frac{\partial}{\partial t_2}\,\mathrm{A}(t_2,\,t_3)) - \mathrm{A}(t_2,\,t_3) + \frac{1}{4}\,\mathrm{A}(t_2,\,t_3)^3 = 0\right\}$$

```
> 2*diff(F(t),t) - F(t) + 1/4*F(t)^3 = 0;
```

$$2\,(\frac{\partial}{\partial t}\,\mathrm{F}(t)) - \mathrm{F}(t) + \frac{1}{4}\,\mathrm{F}(t)^3 = 0$$

```
> simplify( [dsolve( ", F(t), 'explicit' )], symbolic );
```

$$\left[\mathrm{F}(t) = 2\,\frac{e^{(1/2\,t)}}{\sqrt{e^t + 4_C1}},\ \mathrm{F}(t) = -2\,\frac{e^{(1/2\,t)}}{\sqrt{e^t + 4_C1}}\right]$$

```
> simplify( subs( _C1 = 1/4*exp(t)*C, " ), symbolic ):
> subs( C=C*exp(-t), " );
```

$$\left[\mathrm{F}(t) = \frac{2}{\sqrt{1 + C\,e^{(-t)}}},\ \mathrm{F}(t) = -\frac{2}{\sqrt{1 + C\,e^{(-t)}}}\right]$$

Tehát $A(t_2, t_3)$-nak vehető az

$$A(t_2, t_3) = \frac{-2}{\sqrt{1 + C(t_3)e^{-t_2}}}$$

függvény, továbbá föltételezhető, hogy $B(t_2, t_3) = B(t_3)$ alakú. Ekkor az y_1 által kielégített differenciálegyenlet így alakul:

```
> simplify( ode[1], restrictions, convert(
>   [D[1](B)(t2,t3),D[1](A)(t2,t3), A(t2,t3)], diff ) ):
> combine( ", 'trig' );
```

$$(\frac{\partial^2}{\partial t_1{}^2}\,y_1) + y_1 - \frac{1}{4}\,\mathrm{A}(t_2,\,t_3)^3\cos(3\,t_1 + 3\,\mathrm{B}(t_2,\,t_3)) = 0$$

Most egy trükkös számítási lépés következik: a homogén egyenlet megoldásának meghatározása helyett veszünk egy egyszerű alakú y_1 partikuláris megoldást.

```
> eta[1] := (t1,t2,t3) -> 1/32 * A(t2,t3)^3
>   * sin(3*t1+3*B(t2,t3)+3/2*Pi);
```

$$\eta_1 := (t1,\,t2,\,t3) \to \frac{1}{32}\,\mathrm{A}(t2,\,t3)^3\sin(3\,t1 + 3\,\mathrm{B}(t2,\,t3) + \frac{3}{2}\,\pi)$$

```
> ""; # verify the solution
```

$$0 = 0$$

Ezután y_1-et behelyettesítjük a harmadik differenciálegyenletbe, figyelembe véve, hogy B nem függ t_2-től.

```
> eta[1] := 'eta[1]':  eta[0] := 'eta[0]':
> ode[2] := convert( ode[2](t1,t2,t3), diff ):
> eta[1]:= (t1,t2,t3)-> 1/32 * A(t2,t3)^3
>   * sin(3*t1+3*B(t3)+3/2*Pi);
```

$$\eta_1 := (t1,\,t2,\,t3) \to \frac{1}{32}\,\mathrm{A}(t2,\,t3)^3\sin(3\,t1 + 3\,\mathrm{B}(t3) + \frac{3}{2}\,\pi)$$

```
> eta[0] := (t1,t2,t3) -> A(t2,t3) * sin(t1 + B(t3));
```

$$\eta_0 := (t1,\, t2,\, t3) \to \mathrm{A}(t2,\, t3)\sin(t1 + \mathrm{B}(t3))$$

```
> combine( ode[2], 'trig' ):
> ode[2] := collect( ", [ sin(t1+B(t3)), cos(t1+B(t3)) ] ):
> conditions := { coeff( lhs(ode[2]), cos(t1+B(t3)) ) = 0,
>   coeff( lhs(ode[2]), sin(t1+B(t3)) ) = 0 };
```

$$conditions := \Big\{2\,(\frac{\partial}{\partial t_3}\mathrm{A}(t_2,\, t_3)) = 0,\ \frac{3}{4}\mathrm{A}(t_2,\, t_3)^2\,(\frac{\partial}{\partial t_2}\mathrm{A}(t_2,\, t_3)) - (\frac{\partial}{\partial t_2}\mathrm{A}(t_2,\, t_3)) + (\frac{\partial^2}{\partial {t_2}^2}\mathrm{A}(t_2,\, t_3)) - \frac{3}{128}\mathrm{A}(t_2,\, t_3)^5 - 2\,\mathrm{A}(t_2,\, t_3)\,(\frac{\partial}{\partial t_3}\mathrm{B}(t_3)) = 0\Big\}$$

Az egyik föltétel azt jelenti, hogy $A(t_2, t_3)$ nem függ t_3-tól (vagyis $C(t_3)$ konstans). A korábbi kikötésekből levezethetünk $\dfrac{\partial^2 A(t_2,t_3)}{\partial t_2^2}$-re egy szükséges föltételt.

```
> op( remove( has,restrictions,B) );
```

$$2\,(\frac{\partial}{\partial t_2}\mathrm{A}(t_2,\, t_3)) - \mathrm{A}(t_2,\, t_3) + \frac{1}{4}\mathrm{A}(t_2,\, t_3)^3 = 0$$

```
> diff(",t[2]);
```

$$2\,(\frac{\partial^2}{\partial {t_2}^2}\mathrm{A}(t_2,\, t_3)) - (\frac{\partial}{\partial t_2}\mathrm{A}(t_2,\, t_3)) + \frac{3}{4}\mathrm{A}(t_2,\, t_3)^2\,(\frac{\partial}{\partial t_2}\mathrm{A}(t_2,\, t_3)) = 0$$

Az utolsó előtti kikötést is fölhasználva átírhatjuk az előző föltételeket.

```
> simplify( conditions, { ", "" }, convert(
>   [ D[1,1](A)(t2,t3), D[1](A)(t2,t3),
>   A(t2,t3) ], diff ) );
```

$$\Big\{(-2\,(\frac{\partial}{\partial t_3}\mathrm{B}(t_3)) - \frac{1}{4})\,\mathrm{A}(t_2,\, t_3) + \frac{1}{4}\mathrm{A}(t_2,\, t_3)^3 - \frac{9}{128}\mathrm{A}(t_2,\, t_3)^5 = 0,\ 2\,(\frac{\partial}{\partial t_3}\mathrm{A}(t_2,\, t_3)) = 0\Big\}$$

```
> op( select( has, ", B ) );
```

$$(-2\,(\frac{\partial}{\partial t_3}\mathrm{B}(t_3)) - \frac{1}{4})\,\mathrm{A}(t_2,\, t_3) + \frac{1}{4}\mathrm{A}(t_2,\, t_3)^3 - \frac{9}{128}\mathrm{A}(t_2,\, t_3)^5 = 0$$

```
> 'diff(B(t3),t3)' = solve(",diff(B(t3),t3) );
```

$$\frac{\partial}{\partial t_3}\mathrm{B}(t_3) = -\frac{1}{8} + \frac{1}{8}\mathrm{A}(t_2,\, t_3)^2 - \frac{9}{256}\mathrm{A}(t_2,\, t_3)^4$$

Mivel $A(t_2, t_3)$ valójában nem függ t_3-tól,

$$B(t_3) = -\frac{1}{8}\Big(1 - A(t_2, t_3) + \frac{7}{32}A(t_2, t_3)^4\Big)t_3 + B_0,$$

ahol B_0 konstans. Szigorúan véve a $B(t_3)$-ra kapott formula ellentmond annak a korábbi kikötésnek, hogy B nem függ t_2-től, hiszen A *függ* t_2-től. De A képletéből látható, hogy csak lassan változó függvénye t-nek, ezért nem túl nagy t értékekre föltételezhető, hogy A konstans. Az érdeklődő Olvasó a [145]-ben talál ezen ellentmondás föloldásával kapcsolatos részletesebb fejtegetéseket.

Lássuk, milyen közelítést kapunk az $y(0) = 0$, $y'(0) = -0.1$ kezdeti értékekre, ha az $\epsilon = 1$ érték mellett csak $y = y_0 + \epsilon y_1$ alakú approximációt veszünk.

```
> restart: alias( e=epsilon ):
> y := eta[0] + e*eta[1];
```

$$y := \eta_0 + e\,\eta_1$$

```
> eta[0] := t -> A(t)*sin(t+B(t));
```

$$\eta_0 := t \to \mathrm{A}(t)\sin(t + \mathrm{B}(t))$$

```
> eta[1] := t -> -1/32*A(t)^3*cos(3*t+3*B(t));
```

$$\eta_1 := t \to -\frac{1}{32}\,\mathrm{A}(t)^3\cos(3\,t + 3\,\mathrm{B}(t))$$

```
> B := t-> -1/8*(1-A(t)^2+7/32*A(t)^4)*e^2*t+b;
```

$$B := t \to -\frac{1}{8}\,(1 - \mathrm{A}(t)^2 + \frac{7}{32}\,\mathrm{A}(t)^4)\,e^2\,t + b$$

```
> A := t-> -2/(1+c*exp(-e*t));
```

$$A := t \to -\frac{2}{1 + c\,e^{(-e\,t)}}$$

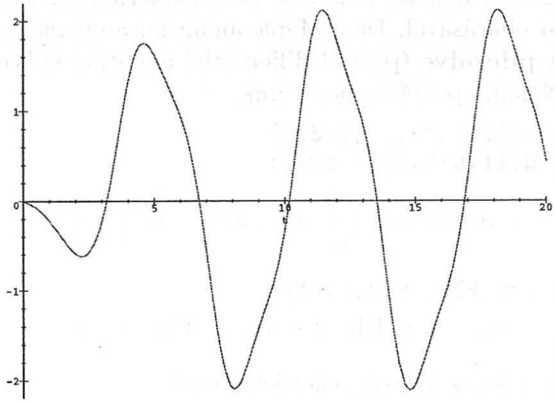
```
> e := 1:  y := unapply( y(t), t ):
> fsolve( { y(0)=0, D(y)(0)=-0.1 }, {b,c},
> {b=-0.1..0.1} );
```

$$\{b = .0004073638406,\ c = 16.51715658\}$$

```
> assign(");
```

Rajzoltassuk is föl ezt a függvényt (17.18. ábra).

```
> plot( y(t), t=0..20 );
```



17.18. ábra: A van der Pol egyenlet megoldása a többszörös skálázásos módszerrel

17.8. Parciális differenciálegyenletek

Az x és t független változóktól függő u ismeretlen függvényre fölírt parciális differenciálegyenlet (PDE) legáltalánosabb alakja

$$F(x, t, u, u_x, u_t, u_{xx}, u_{xt}, u_{tt}, \ldots) = 0,$$

ahol az indexek a megfelelő változók szerinti parciális differenciálást jelentenek, tehát

$$u_x = \frac{\partial u}{\partial x}, \; u_{xt} = \frac{\partial^2 u}{\partial x \partial t}, \; u_{tt} = \frac{\partial^2 u}{\partial t^2}, \; \ldots$$

A PDE *rendjét* a KDE-hez hasonlóan a PDE-ben előforduló legmagasabb rendű deriválttal határozzuk meg. A legáltalánosabb elsőrendű PDE tehát az x, t, u, u_x és u_t változókkal

$$F(x, t, u, u_x, u_t) = 0$$

alakban írható föl. Ha F a benne előforduló legmagasabb rendű parciális derivált r-ed fokú polinomja, akkor azt mondjuk, hogy a PDE *foka r*. Ha speciálisan F lineáris, akkor a PDE-t is *lineárisnak* hívjuk. *Kvázilineárisnak* nevezzük az olyan egyenletet, amely a legmagasabb rendű parciális deriváltban lineáris, de tartalmaz egyéb nemlineáris tagokat. Néhány ismert példa:

$$u_t - u_{xx} = 0$$ hővezetési vagy diffúziós egyenlet: másodrendű lineáris PDE

$$u_{tt} - u_{xx} = 0$$ hullámegyenlet: másodrendű lineáris PDE

$$u_t - u\, u_x + u_{xxx} = 0$$ Korteweg–de Vries egyenlet: harmadrendű elsőfokú kvázilineáris PDE

Korábban már láttuk, hogyan oldhatók meg numerikusan PDE-k a `DEtools` csomag **PDEplot** eljárásával. De a Maple analitikusan is meg tud oldani bizonyos PDE-ket. A **pdesolve** (partial differential equation solver) eljárás végzi ezt a feladatot. Néhány példát ismertetünk:

```
> wavePDE := diff(u(x,t),[t$2]) -
>   1/v^2 * diff(u(x,t),[x$2]);
```

$$wavePDE := (\frac{\partial^2}{\partial t^2}\mathrm{u}(x,\, t)) - \frac{\frac{\partial^2}{\partial x^2}\mathrm{u}(x,\, t)}{v^2}$$

```
> pdesolve( wavePDE, u(x,t) );
```

$$\mathrm{u}(x,\, t) = _\mathrm{F1}(-t - v\,x) + _\mathrm{F2}(t - v\,x)$$

Itt `_F1` és `_F2` tetszőleges függvényeket jelentenek.

```
> PDE := a*x*diff(u(x,t),x) + b*t*diff(u(x,t),t) = 0;
```

$$PDE := a\,x\,(\frac{\partial}{\partial x}\mathrm{u}(x,\, t)) + b\,t\,(\frac{\partial}{\partial t}\mathrm{u}(x,\, t)) = 0$$

```
> pdesolve( ", u(x,t) );
```

$$\mathrm{u}(x,\, t) = _\mathrm{F1}\left(\frac{t}{x^{(\frac{b}{a})}}\right)$$

```
> PDE := a*x^2*diff(u(x,t),x) + b*t^2*diff(u(x,t),t) = 0;
```

$$PDE := a\,x^2\,(\frac{\partial}{\partial x}\mathrm{u}(x,\, t)) + b\,t^2\,(\frac{\partial}{\partial t}\mathrm{u}(x,\, t)) = 0$$

```
> pdesolve( PDE, u(x,t) );
```

$$\mathrm{u}(x,\, t) = _\mathrm{F1}(\frac{-b\,t + a\,x}{t\,a\,x})$$

```
> PDE := diff(u(x,t),x$2) - t^2*diff(u(x,t),t$2)
>   - t*diff(u(x,t),t)=0;
```

$$PDE := (\frac{\partial^2}{\partial x^2}\mathrm{u}(x,\, t)) - t^2\,(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,\, t)) - t\,(\frac{\partial}{\partial t}\mathrm{u}(x,\, t)) = 0$$

```
> pdesolve( PDE, u(x,t) );
```

$$\mathrm{u}(x,\, t) = _\mathrm{F1}(t\,e^x) + _\mathrm{F2}(\frac{t}{e^x})$$

Ne adjuk föl túl korán, ha a Maple-től nem kapunk határozott választ. Nézzük például a hővezetés egyenletét:

```
> heatPDE := diff(u(x,t),t) = diff(u(x,t),[x$2]);
```

$$heatPDE := \frac{\partial}{\partial t}\mathrm{u}(x,\, t) = \frac{\partial^2}{\partial x^2}\mathrm{u}(x,\, t)$$

```
> pdesolve( heatPDE, u(x,t) );
```

$$\mathrm{pdesolve}(\frac{\partial}{\partial t}\mathrm{u}(x,\, t) = \frac{\partial^2}{\partial x^2}\mathrm{u}(x,\, t),\, \mathrm{u}(x,\, t))$$

A rendszer nem talált megoldást. Alkalmazzunk Fourier-transzformációt a helykoordinátára vonatkozóan:

```
> with(inttrans): # load library package
> diffeqn := fourier( heatPDE, x, w );
```

$$diffeqn := \frac{\partial}{\partial t}\,\mathrm{fourier}(\mathrm{u}(x,\,t),\,x,\,w) = -w^2\,\mathrm{fourier}(\mathrm{u}(x,\,t),\,x,\,w)$$

Vegyünk föl $u(x, 0) = T\delta(x)$ alakú kezdeti föltételt (ennek ismerjük a Fourier-transzformáltját). Ekkor a következő közönséges differenciálegyenletet kell megoldanunk:

```
> ODE := subs( fourier(u(x, t), x, w) = U(t), diffeqn );
```

$$ODE := \frac{\partial}{\partial t}\,\mathrm{U}(t) = -w^2\,\mathrm{U}(t)$$

A hozzá tartozó kezdeti föltétel pedig

```
> initval := U(0) = fourier( T*Dirac(x), x, w );
```

$$inital := \mathrm{U}(0) = T$$

Ezt a kezdetiérték problémát már gond nélkül megoldja a Maple.

```
> dsolve( { ODE, initval }, U(t) );
```

$$\mathrm{U}(t) = e^{(-w^2\,t)}\,T$$

Az eredményt visszatranszformáljuk, föltéve, hogy $t > 0$.

```
> assume(t>0):
> u(x,t) = invfourier( rhs("), w, x );
```

$$\mathrm{u}(x,\,t\tilde{}) = \frac{1}{2}\,\frac{T\,\sqrt{\dfrac{\pi}{t\tilde{}}}\,e^{(-1/4\,\frac{x^2}{t\tilde{}})}}{\pi}$$

```
> simplify(");
```

$$\mathrm{u}(x,\,t\tilde{}) = \frac{1}{2}\,\frac{T\,e^{(-1/4\,\frac{x^2}{t\tilde{}})}}{\sqrt{\pi}\,\sqrt{t\tilde{}}}$$

A Maple-nek nyújtott kis segítséggel tehát tudtunk találni egy megoldást a kezdetiérték problémára. A változók szétválasztásának módszere a másik olyan példa, ahol kis támogatással meg tudja határozni a számítógépes algebrai rendszer a peremérték-problémák megoldásait.

17.9. Parciális differenciálegyenletek Lie szimmetriái

A Lie szimmetriák módszere a differenciálegyenletek tanulmányozásánál fölhasználható egyik leghasznosabb technika. A Maple `liesymm` csomagjában találjuk a Lie szimmetria módszer Harrison és Estabrook által kidolgozott formalizmusának (lásd [35, 96]) alkalmazásához szükséges eszközöket. Ebben a részben a csomag segítségével az

$$u_t + u\,u_x + u_{xxx} = 0$$

Korteweg–de Vries egyenlet Lie szimmetriáit határozzuk meg:

```
> with(liesymm): # load library package
> KdV_eqn := Diff(u(t,x),t) + u(t,x)*Diff(u(t,x),x)
>   + Diff(u(t,x),[x$3])=0;
```

$$KdV_eqn := (\frac{\partial}{\partial t}\,\mathrm{u}(t,\,x)) + \mathrm{u}(t,\,x)\,(\frac{\partial}{\partial x}\,\mathrm{u}(t,\,x)) + (\frac{\partial}{\partial x^3}\,\mathrm{u}(t,\,x)) = 0$$

A parciális differenciálegyenletekre vonatkozó Lie szimmetria módszerek alapgondolata a következő (v. ö. [149, 166, 172]). Az

$$\omega(t, x, u, u_x, u_t, u_{xx}, u_{xt}, u_{tt}, \ldots) = 0$$

PDE *Lie szimmetriája* olyan

$$t \to \bar{t}(t,x,u), \qquad x \to \bar{x}(t,x,u), \qquad u \to \bar{(u)}(t,x,u)$$

leképezést jelent, amellyel bevezetett új változók szintén kielégítik az eredeti egyenletet. Általában csak Lie szimmetriák egyparaméteres csoportjait szokás tekinteni. Ebben az esetben az infinitézimális leképezések a következők:

$$t \to t + \epsilon\tau, \qquad x \to x + \epsilon\xi, \qquad u \to u + \epsilon\eta.$$

A PDE változatlan marad, ha teljesül az

$$X\omega = 0$$

egyenlet, ahol az X operátor definíciója

$$X = \tau\partial_t + \xi\partial_x + \eta\partial_u + \cdots.$$

Ebből a tulajdonságból a τ, ξ és η ismeretlenekre vonatkozó lineáris homogén parciális differenciálegyenletekből álló rendszert kapunk, amelyet determináló rendszernek („determining system") fogunk nevezni. A Maple **determine** eljárásával íratható föl a rendszer:

```
> eqns[1] := determine( KdV_eqn, V, u(t,x), w );
```

$$eqns_1 := \{\frac{\partial^3}{\partial u^3}\,\mathrm{V2}(t,\,x,\,u) = 0,\ \frac{\partial}{\partial t}\,\mathrm{V2}(t,\,x,\,u) = 2\,u\,(\frac{\partial}{\partial x}\,\mathrm{V2}(t,\,x,\,u))$$
$$-\,(\frac{\partial^3}{\partial x^3}\,\mathrm{V2}(t,\,x,\,u)) + \mathrm{V3}(t,\,x,\,u) + 3\,(\frac{\partial^3}{\partial x^2\,\partial u}\,\mathrm{V3}(t,\,x,\,u)),$$
$$\frac{\partial^3}{\partial x^3}\,\mathrm{V3}(t,\,x,\,u) = -u\,(\frac{\partial}{\partial x}\mathrm{V3}(t,\,x,\,u)) - (\frac{\partial}{\partial t}\mathrm{V3}(t,\,x,\,u)),$$
$$\frac{\partial^2}{\partial u^2}\,\mathrm{V2}(t,\,x,\,u) = 0,\ \frac{\partial}{\partial u}\,\mathrm{V1}(t,\,x,\,u) = 0,\ \frac{\partial}{\partial x}\,\mathrm{V1}(t,\,x,\,u) = 0,$$
$$\frac{\partial^3}{\partial x^2\,\partial u}\,\mathrm{V1}(t,\,x,\,u) = 0,\ \frac{\partial^2}{\partial u^2}\,\mathrm{V3}(t,\,x,\,u) = 3\,(\frac{\partial^2}{\partial x\,\partial u}\,\mathrm{V2}(t,\,x,\,u)),$$
$$\frac{\partial^3}{\partial x^2\,\partial u}\,\mathrm{V2}(t,\,x,\,u) = \frac{\partial^3}{\partial x\,\partial u^2}\,\mathrm{V3}(t,\,x,\,u),$$
$$\frac{\partial^3}{\partial u^3}\,\mathrm{V3}(t,\,x,\,u) = 3\,(\frac{\partial^3}{\partial x\,\partial u^2}\,\mathrm{V2}(t,\,x,\,u)),$$

$$\frac{\partial}{\partial t}\mathrm{V1}(t,\,x,\,u)=3\,(\frac{\partial}{\partial x}\mathrm{V2}(t,\,x,\,u))-(\frac{\partial^3}{\partial x^3}\mathrm{V1}(t,\,x,\,u)),$$
$$\frac{\partial^3}{\partial x\,\partial u^2}\mathrm{V1}(t,\,x,\,u)=0,\ \frac{\partial^2}{\partial x\,\partial u}\mathrm{V1}(t,\,x,\,u)=0,\ \frac{\partial^2}{\partial x^2}\mathrm{V1}(t,\,x,\,u)=0,$$
$$\frac{\partial}{\partial u}\mathrm{V2}(t,\,x,\,u)=0,\ \frac{\partial^2}{\partial u^2}\mathrm{V1}(t,\,x,\,u)=0,\ \frac{\partial^3}{\partial u^3}\mathrm{V1}(t,\,x,\,u)=0,$$
$$\frac{\partial^2}{\partial x^2}\mathrm{V2}(t,\,x,\,u)=\frac{\partial^2}{\partial x\,\partial u}\mathrm{V3}(t,\,x,\,u)\}$$

Itt néhány jelölésbeli változtatást alkalmaztunk:

$$V1=\tau,\quad V2=\xi,\quad V3=\eta.$$

A determináló rendszert a Maple **autosimp** eljárásával egyszerűbb alakra hozhatjuk, sőt, szerencsés esetben akár meg is oldhatjuk. Lássuk, mi történik a jelenlegi esetben:

```
> eqns[2] := autosimp( eqns[1] );
```

$$eqns_2 := \{\}\ \&\mathrm{where}\{\mathrm{V2_3}(t)=t\,C5+C6,\ \mathrm{V2_2}(t)=t\,C7+C8,$$
$$\mathrm{V1}(t,\,x,\,u)=C9+\frac{3}{2}t^2\,C7+3\,C8\,t,\ \mathrm{V1_2}(t)=C9+\frac{3}{2}t^2\,C7+3\,C8\,t,$$
$$\mathrm{V3_2}(t,\,x)=x\,C7+C5,\ \mathrm{V3_4}(t)=C7,\ \mathrm{V3_5}(t)=C5,$$
$$\mathrm{V3_3}(t)=-t\,C7+C4,\ \mathrm{V3}(t,\,x,\,u)=u\,(-t\,C7+C4)+x\,C7+C5,$$
$$-t\,C7-2\,C8-C4=0,\ \mathrm{V3_1}(t,\,x)=-t\,C7+C4,$$
$$\mathrm{V2_1}(t,\,x)=x\,(t\,C7+C8)+t\,C5+C6,$$
$$\mathrm{V2}(t,\,x,\,u)=x\,(t\,C7+C8)+t\,C5+C6\}$$

A Maple megoldotta a determináló rendszert. A megoldás azonban tartalmazza a $tC7-C4-2C8=0$ egyenletet, amely minden t-re csak úgy teljesülhet, ha $C7=0$ és $C4=-2C8$. Ha elvégezzük ezeket a helyettesítéseket, $V1$, $V2$ és $V3$-ra a következő kifejezéseket kapjuk:

```
> eqns := subs( C3=0,C6=-2*C4, op(2,eqns[2]) ):
> select( has, eqns, {V1,V2,V3} );
```

$$\{\mathrm{V1}(t,\,x,\,u)=C9+\frac{3}{2}t^2\,C7+3\,C8\,t,$$
$$\mathrm{V2}(t,\,x,\,u)=x\,(t\,C7+C8)+t\,C5-2\,C4,$$
$$\mathrm{V3}(t,\,x,\,u)=u\,(-t\,C7+C4)+x\,C7+C5\}$$

Olyan szerencsések voltunk, hogy megtaláltuk az általános megoldást:

$$\begin{aligned}\tau &= c_9+3c_8t,\\ \xi &= c_6+c_5t+c_8x,\\ \eta &= c_5-2c_8u.\end{aligned}$$

Itt c_5, c_6, c_8 és c_9 tetszőleges konstansok. A Korteweg–de Vries (Kdv) egyenlet négydimenziós szimmetria algebráját tehát az alábbi táblázatban fölsorolt szimmetriák feszítik ki:

Szimmetria	**Jelentése**
∂_t	idő eltolás
∂_x	hely eltolás
$t\partial_x + \partial_u$	Galilei-gyorsítás
$x\partial_x + 3t\partial_t - 2u\partial_u$	skálázás

17.5. táblázat: A KdV egyenlet Lie szimmetriái

A fönti Lie szimmetriák előnye abban mutatkozik meg, hogy ha $u = f(t, x)$ a Korteweg–de Vries egyenlet megoldása, akkor

$$\begin{aligned} u_1 &= f(t-\epsilon, x), \\ u_2 &= f(t, x-\epsilon), \\ u_3 &= f(t, x-\epsilon t)+\epsilon, \text{ és} \\ u_4 &= e^{-2\epsilon} f(e^{-3\epsilon}t, e^{-\epsilon}x) \end{aligned}$$

is megoldás minden valós ϵ-ra.

Előző sikereink ellenére meg kell jegyeznünk, hogy a `liesymm` csomagnak a determináló rendszer automatikus megoldásával kapcsolatos képességei a jelenlegi implementációban még eléggé korlátozottak. Ezért néha sok manuális számolásra van szükség. A [101] irodalom érdekes összeállítást tartalmaz a parciális differenciálegyenletek szimmetriáinak meghatározására alkalmazható egyéb csomagokról.

17.10. Gyakorlatok

1. Mutassuk meg, hogy

$$\sqrt{a^2-y^2} - a\ln(a+\sqrt{a^2-y^2}) + a\ln y + x = C$$

az

$$y' = \frac{y}{\sqrt{a^2-y^2}}$$

differenciálegyenlet olyan implicit megoldása, amely az x-tengely mentén jobbra mozgó személy által a hosszúságú kötélen vontatott objektum pályáját írja le.

1. Oldjuk meg a következő közönséges differenciálegyenleteket a Maple segítségével. Próbáljunk ki különböző módszereket, igyekezzünk a megoldások legegyszerűbb alakját megkeresni. Ellenőrizzük, hogy a Maple megtalálta-e az összes megoldást.

(a) $3y^2y' + 16x = 12xy^3$.

(b) $y' = 2\frac{y}{x} - \left(\frac{y}{x}\right)^2$.

(c) $xy' - y = x \tan\left(\frac{y}{x}\right)$.

3. Néhány egynél magasabb fokú közönséges differenciálegyenlet is megoldható a Maple-lel. Tekintsük a következő két példát. Oldjuk meg az

(a) $y'^2 - y^2 = 0$ és az

(b) $y'^2 + xy = y^2 + xy'$ egyenletet.

Meg tudjuk-e mondani ezek alapján, hogy a Maple hogyan próbál megbirkózni a magasabb fokú differenciálegyenletekkel?

4. Oldjuk meg a Maple-lel a következő KDE-ket; ellenőrizük az eredményeket:

(a) $x^4y'' - (2x^2 - 1)xy' + y = 0$.

(b) $x^2y'' + 3xy' + (x^2 - 35)y = x$.

(c) $y' + xy^2 = 1$.

5. Tekintsük a következő kezdetiérték problémát:

$$y'' - y = 0, \quad y(0) = 1, \quad y'(0) = 0.$$

(a) Keressük meg a megoldást Laplace transzformációval.

(b) Végezzük el újra az (a)-beli számításokat a `printlevel` változó értékét 3-ra változtatva.

(c) Ismételjük meg (a)-t, de most a `printlevel` változót állítsuk 33-ra.

6. Számítsuk ki a következő kezdetiérték probléma Taylor-sorral fölírt megoldásának első 10 tagját:

$$y' = yz, \quad z' = xz + y, \quad y(0) = 1, \quad z(0) = 0.$$

7. Tekintsük az Airy-féle differenciálegyenletet:

$$y'' + xy = 0.$$

(a) Keressük meg az $y(0) = 1$ és $y'(0) = 0$ kezdeti értékeknek megfelelő megoldást a hatványsor módszer segítségével. Milyen 30-nál kisebb fokszámú tagok szerepelnek a megoldásban?

(b) A feladat (a) részében talált hatványsor együtthatói között milyen rekurrens reláció teljesül?

(c) Határozzuk meg a megoldást az $y(0) = 0$ és $y'(0) = 1$ kezdeti értékekre hatványsor módszerrel. Milyen 30-nál alacsonyabb fokszámú tagok szerepelnek a megoldásban?

8. Tekintsük a Duffing-féle differenciálegyenletet:

$$x'' + x + \epsilon x^3 = \epsilon F \cos \omega t.$$

Alkalmazzuk a Poincaré–Lindstedt módszert a KDE azon periodikus megoldásának közelítésére, amely kielégíti az

$$x(0) = A \qquad x'(0) = 0$$

kezdeti értékeket. Milyen eredményt kapunk, ha a többszörös skálázás módszerét alkalmazzuk, és hogyan viszonyul ez a két módszer a numerikus közelítéshez?

9. Határozzuk meg a Burgers-féle

$$u_t + u_{xx} + 2u\, u_x = 0$$

egyenlet determináló rendszerét, és számítsuk ki a szimmetria algebráját.

10. Számítsuk ki az

$$u_{tx} + u_x + u^2 = 0$$

Boltzann-egyenlet determináló rendszerét, és próbáljuk meg kiszámítani a szimmetria algebráját. (Útmutatás: szükségünk lehet a **pdint** eljárás betöltésére a `liessymm/difftools` „rejtett” csomagból.)

18.

Lineáris algebra: a `linalg` csomag

A fejezetben a Maple mátrixokkal kapcsolatos számításokban fölhasználható alapvető eszközeit vizsgáljuk. Az olyan elemi műveleteken túl, mint az összeadás vagy a szorzás, szó lesz „magasabb szintű" számításokról, determinánsokról, invertálásról, sajátértékekről és sajátvektorokról is. Áttekintjük a `linalg` lineáris algebra csomag szolgáltatásait. A vektorok és mátrixok speciális kiértékelési szabályaival kapcsolatban a 12.6. alfejezetre utalunk.

18.1. A `linalg` csomag betöltése

Mielőtt a vektorokkal és a mátrixokkal számolni kezdenénk, célszerű betölteni a `linalg` csomagot, amelyet kimondottan lineáris algebrai számításokhoz állítottak össze.

```
> with(linalg);

Warning, new definition for norm

Warning, new definition for trace
```

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *LUdecomp*, *QRdecomp*, *Wronskian*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *cholesky*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvalues*, *eigenvectors*,

eigenvects, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *forwardsub*, *frobenius*, *gausselim*, *gaussjord*, *geneqns*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *issimilar*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matadd*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stack*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*, *wronskian*]

A csomag betöltése utáni figyelmeztetés arra emlékeztet bennünket, hogy a **norm** és a **trace** függvények már korábban is léteztek a Maple-ben, de most a rendszer a `linalg` csomag azonos nevű függvényeivel helyettesítette őket. Valamely eljárás eredeti jelentéséhez úgy térhetünk vissza, ha explicit módon újra betöltjük. Például a **norm** függvény eredeti definícióját (ti. polinomok normája) a következő parancs után érhetjük el:

```
> readlib( norm );
```

proc(p, n, v) ... **end**

Töltsük be újra a **linalg[norm]** eljárást.

```
> with( linalg, norm );

Warning, new definition for norm
```

$[norm]$

A rendelkezésünkre álló függvények fönti listájából világos, hogy elérhető a legtöbb ismert mátrixművelet: mátrixok szorzása és összeadása, sor- és oszlopműveletek, Gauss-elimináció és fölső trianguláris alak, mátrixok determinánsa, nyoma, inverze, karakterisztikus polinomja, karakterisztikus mátrixa, sajátértékei és sajátvektorai stb. A `linalg` egyes függvényeiről az online help nyújt részletes tájékoztatást. Ebben a fejezetben egy csomó olyan példát adunk meg, melyek a leggyakoribb mátrixokkal végzett számításokkal kapcsolatosak. Még több példát közöl [110]. A következő fejezetben a lineáris algebra bonyolultabb alkalmazásait tekintjük. A `linalg` számos lehetősége, ezek hasznossága azonban leginkább sok gyakorlással ismerhető meg.

A fejezet során végig föltesszük, hogy az Olvasó betöltötte a `linalg` *csomagot.*

18.2. Új vektorok és mátrixok létrehozása

A 12.4. alfejezetben leírtak szerint a Maple-ben a vektorokat és a mátrixokat *tömbökként* implementálták, ezért az **array** eljárás hívásával hozhatók létre. Eközben használhatunk olyan indexfüggvényeket is, mint a *symmetric* vagy a

sparse. Alternatív megoldásként használhatjuk a **vector** és a **matrix** rutinokat is. A vektorok és mátrixok ily módon történő létrehozása különösen akkor kényelmes, ha valamely indexfüggvény segítségével rögtön meg is tudjuk adni az összes elemet. Néhány példa következik.

```
> A := array( 1..2, 1..3, [[a,b,c],[d,e,f]] );
```

$$A := \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

```
> B := array( sparse, antisymmetric, 1..2, 1..2 ):
> B[1,2] := x:  print(B);
```

$$\begin{bmatrix} 0 & x \\ -x & 0 \end{bmatrix}$$

```
> v := vector( [1,2,3] );
```

$$v := [1, 2, 3]$$

```
> C := matrix( 4, 4, (i,j) -> i^(j-1) );
```

$$C := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$$

A C mátrix a Vandermonde mátrixok speciális esete; erről a **vandermonde** beépített függvény használatával is meggyőződhetünk.

```
> vandermonde( [1,2,3,4] );
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$$

A Maple speciális mátrixokkal kapcsolatos további beépített rutinjait a 18.1. táblázatban soroltuk föl. Részletesebb információkat az online help-ből kaphatunk.

Az **entermatrix** eljárás a promptjel ismételt kiírásával egyenként kéri a mátrixelemeket.

```
> M := array( antisymmetric, 1..3, 1..3 ):
> entermatrix( M );

enter element 1,2 > alpha;
enter element 1,3 > beta;
enter element 2,3 > gamma;
```

$$\begin{bmatrix} 0 & \alpha & \beta \\ -\alpha & 0 & \gamma \\ -\beta & -\gamma & 0 \end{bmatrix}$$

Eljárás	*Mátrix*
JordanBlock	Jordan normálforma
bezout	két polinom Bezout mátrixa
companion	polinom kísérő mátrixa
fibonacci	Fibonacci mátrix
grad	kifejezés gradiensvektora
hessian	kifejezés Hesse mátrixa
hilbert	általánosított Hilbert mátrix
jacobian	vektorfüggvény Jacobi mátrixa
sylvester	két polinom Sylvester mátrixa
toeplitz	listával megadott szimmetrikus Töplitz mátrix
vandermonde	listával megadott Vandermonde mátrix
wronskian	függvények listájának Wronski mátrixa

18.1. táblázat: Speciális vektorokkal és mátrixokkal kapcsolatos eljárások

hessian@hessian Szalag- és diagonális mátrixok a **band**, illetve a **diag** eljárással hozhatók létre.

```
> band( [-1,2,1], 4 );
```

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ -1 & 2 & 1 & 0 \\ 0 & -1 & 2 & 1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

```
> diag( mu, nu );
```

$$\begin{bmatrix} \mu & 0 \\ 0 & \nu \end{bmatrix}$$

```
> diag( "", " );
```

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \nu \end{bmatrix}$$

Az utolsó példából az is látható, hogyan hozhatunk létre korábban definiált mátrixokból álló blokkos mátrixot a **diag** segítségével.

A **genmatrix** olyan rutinra példa, amely más adatstruktúrából készít mátrixot: a megadott lineáris egyenletrendszerhez tartozó együtthatómátrixot generálja.

```
> eqns := { (c^2-1)*x + (c-1)*y = (1-c)^2,
>   (c-1)*x + (c^2-1)*y = c-1 };
```

$$eqns := \{(c^2-1)\,x + (c-1)\,y = (1-c)^2,\ (c-1)\,x + (c^2-1)\,y = c-1\}$$

```
> genmatrix( eqns, [x,y], 'flag' );
```

$$\begin{bmatrix} c^2-1 & c-1 & (1-c)^2 \\ c-1 & c^2-1 & c-1 \end{bmatrix}$$

Ezután mátrixalgebrai eszközökkel megoldhatjuk a rendszert. De miért nem a **solve**-val próbálkozunk? Lássuk, mi történik, ha egyenletrendszerre alkalmazzuk az eljárást.

```
> solve( eqns, {x,y} );
```

$$\left\{y = \frac{2}{c\,(c+2)},\ x = \frac{c^2-2}{c\,(c+2)}\right\}$$

Ez egy megoldás! De ha alaposabban megnézzük a rendszert, észrevehetjük, hogy $c = 1$-re bármely (x, y) megoldás, továbbá a $c = 0$ vagy a $c = -2$ esetben nincs megoldás. Ha kiszámítjuk a mátrix Hermite normálformáját ($\mathbb{Q}(c)$ fölötti fölső trianguláris alakját), nyilvánvalóvá válnak az előbb említett speciális esetek:

```
> genmatrix( eqns, [x,y], 'flag' );
```

$$\begin{bmatrix} c^2-1 & c-1 & (1-c)^2 \\ c-1 & c^2-1 & c-1 \end{bmatrix}$$

```
> hermite( ", c );
```

$$\begin{bmatrix} c-1 & c^2-1 & c-1 \\ 0 & c^3+c^2-2\,c & (c+1)\,(c-1)-1+2\,c-c^2 \end{bmatrix}$$

```
> map( factor, " );
```

$$\begin{bmatrix} c-1 & (c+1)\,(c-1) & c-1 \\ 0 & c\,(c+2)\,(c-1) & 2\,c-2 \end{bmatrix}$$

Folytassuk a vektorok és mátrixok generálásáról szóló áttekintésünket. A **randvector** és a **randmatrix** eljárásokkal véletlen vektorokat, illetve mátrixokat állíthatunk elő. Flexibilis opciók segítségével megadhatjuk a mátrixok formáját és az elemek alakját. A mátrix formája a következőkből választható: ritka, sűrű (ez az alapértelmezés), szimmetrikus, antiszimmetrikus vagy unimoduláris (azaz 1 determinánsú). Ha `entries = f` alakú opcionális argumentum is szerepel, ahol **f** valamely Maple eljárás neve, akkor az elemek generálása **f** hívásával történik. Az alapértelmezés szerinti függvény a **rand(-99..99)**, ekkor kétjegyű véletlen egészeket kapunk. Elég lesz egyetlen példa.

```
> poly := proc() randpoly( x, terms=3, degree=3 ) end:
> randmatrix( 3, 3, entries=poly, unimodular );
```

$$\begin{bmatrix} 1 & -55-37x^3-35x & 50+79x^2+56x \\ 0 & 1 & 63+57x^3-59x \\ 0 & 0 & 1 \end{bmatrix}$$

Végül, de nem utolsósorban fájlból beolvasott adatok fölhasználásával is létrehozhatunk mátrixokat. Tegyük föl, hogy az aktuális könyvtárban található háromsoros `matdata` nevű fájl a következő számokat tartalmazza:

```
1    2    3
4    5    6
7    8    9
```

A **readdata** paranccsal importálhatjuk és mátrix alakban tárolhatjuk ezeket az adatokat:

```
> readlib(readdata):
> mat := matrix( readdata( `matdata`, 3, integer ) );
```

$$mat := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Adatok exportálásáról/importálásáról és az alacsony szintű I/O-ról a 4.4. és a 4.5. alfejezetekben volt szó részletesebben.

18.3. Vektor- és mátrixaritmetika

Demonstrációs célokra fogjuk fölhasználni a következő mátrixokat:

```
> A := matrix( 2, 2, [ a, b, c, d ] );
```

$$A := \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
> B := toeplitz( [ alpha, beta ] );
```

$$B := \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix}$$

```
> C := matrix( 3, 2, (i,j) -> i+j-1 );
```

$$C := \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}$$

Az utolsó névig történő kiértékelés miatt láthatóan nem lehet az

```
> A + B;
```

$$A + B$$

formulával kiszámítani a mátrixok összegét. Ehelyett használhatjuk a **matadd** (**mat**rix **add**ition) függvényt:

```
> matadd( A, B );
```

$$\begin{bmatrix} a+\alpha & b+\beta \\ c+\beta & d+\alpha \end{bmatrix}$$

A mátrixokkal végzett számítások leghasznosabb segédeszköze a Maple **evalm** (**eval**uate using **m**atrix arithmetic) eljárása:

```
> evalm( A + B );
```

$$\begin{bmatrix} a+\alpha & b+\beta \\ c+\beta & d+\alpha \end{bmatrix}$$

```
> evalm( 3*A - 2/7*B );
```

$$\begin{bmatrix} 3a - \frac{2}{7}\alpha & 3b - \frac{2}{7}\beta \\ 3c - \frac{2}{7}\beta & 3d - \frac{2}{7}\alpha \end{bmatrix}$$

Skalárok hozzáadása is lehetséges, ezek a fődiagonális elemeihez adódnak hozzá:

```
> evalm( A - 1 );
```

$$\begin{bmatrix} a-1 & b \\ c & d-1 \end{bmatrix}$$

```
> evalm( C + c );
```

$$\begin{bmatrix} 1+c & 2 \\ 2 & 3+c \\ 3 & 4 \end{bmatrix}$$

Mátrixok szorzására nem használhatjuk a $*$ operátort: ez a Maple-ben mindig a *kommutatív* szorzást jelenti.

```
> A*B + 2*B*A;
```

$$3\,A\,B$$

```
> A*A + B*C*B;
```

$$A^2 + B^2\,C$$

Amint látható, a Maple az automatikus egyszerűsítések során nem veszi figyelembe az A, B és a C nevek típusát. Ha meg az **evalm**-mel próbáljuk a szorzatmátrixot kiszámoltatni, a Maple nem mindig az általunk helyesnek vélt eredményt hozza ki:

```
> evalm( B * A );

Error, (in evalm/evaluate)
use the \&* operator for matrix/vector multiplication
```

Mátrixok szorzatának jelölésére a `&*` szorzásoperátort vagy a **multiply** eljárást kell alkalmazni:

```
> evalm( B &* A );
```

$$\begin{bmatrix} \alpha\,a+\beta\,c & \alpha\,b+\beta\,d \\ \beta\,a+\alpha\,c & \beta\,b+\alpha\,d \end{bmatrix}$$

```
> evalm( A &* B );
```

$$\begin{bmatrix} \alpha\,a+\beta\,b & \beta\,a+\alpha\,b \\ \alpha\,c+\beta\,d & \beta\,c+\alpha\,d \end{bmatrix}$$

A `&*` szorzásoperátor mátrixok vektorral való szorzásánál is alkalmazható. A Maple a vektorokat a szövegkörnyezetnek megfelelően hol sor-, hol oszlopvektoroknak tekinti.

```
> v := vector( [x,y] ); w := vector([xi,eta]);
```

$$v := [x,\, y]$$

$$w := [\xi,\, \eta]$$

```
> evalm( A &* v ); # v used as column vector
```

$$[a\,x + b\,y,\; c\,x + d\,y]$$

```
> evalm( v &* A ); # v used as row vector
```

$$[a\,x + y\,c,\; x\,b + d\,y]$$

```
> evalm( v &* w ); # mixed situation as inner product
```

$$x\,\xi + y\,\eta$$

Ha (például pedagógiai okokból) azt szeretnénk, hogy az oszlopvektorokat valóban oszlopformátumban jelenítse meg a rendszer, a következő szokásos trükköt alkalmazhatjuk. Először is vessünk egy pillantást a vektorok kiírását végző **print/array/vector** rutinra:

```
> interface( verboseproc=2 ):
> print( `print/array/vector` );

proc(A,aname)
local i,n,lA,v;
options `Copyright 1992 by the University Waterloo`;
    i := [ op(2,A) ];
    n := op(2,i[1]);
    lA := A;
    v := [ seq( eval(lA[i],1), i=1..n) ];
    subs( 'lA'=aname, v );
end:
```

Látható, hogy a Maple a megjelenítéshez egy **VECTOR(...)** alakú függvényhívást konstruál meg. Csak annyit kell tennünk, hogy ezt utánozva végeredményként a `column` attributummal ellátott listát hozunk létre. Ettől kezdve a Maple a vektorokat oszlopvektorokként írja ki:

```
> `print/array/vector` := proc(A,aname)
>  local i,n,lA,v,column;
>  i := [ op(2,A) ];
>  n := op(2,i[1]);
>  lA := A;
>  v := [ seq( eval(lA[i],1), i=1..n) ];
>  subs( 'lA'=aname, v );
>  setattribute( v, column )
> end:
> print( v );
```

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

```
> evalm( A &* v );
```

$$\begin{bmatrix} a\,x + b\,y \\ c\,x + d\,y \end{bmatrix}$$

Mátrixok hatványai a szokásos ^ operátorral számíthatók ki:

```
> evalm( B^3 );
```

$$\begin{bmatrix} (\alpha^2+\beta^2)\,\alpha + 2\,\alpha\,\beta^2 & (\alpha^2+\beta^2)\,\beta + 2\,\alpha^2\,\beta \\ (\alpha^2+\beta^2)\,\beta + 2\,\alpha^2\,\beta & (\alpha^2+\beta^2)\,\alpha + 2\,\alpha\,\beta^2 \end{bmatrix}$$

A **map** eljárás lehetővé teszi, hogy egyidejűleg egyszerűsítsük az összes mátrixelemet.

```
> map( factor, " );
```

$$\begin{bmatrix} \alpha\,(\alpha^2+3\,\beta^2) & \beta\,(3\,\alpha^2+\beta^2) \\ \beta\,(3\,\alpha^2+\beta^2) & \alpha\,(\alpha^2+3\,\beta^2) \end{bmatrix}$$

A Maple nem csak természetes számokat fogad el kitevőként: ha a mátrix nem szinguláris, használhatunk negatív egész kitevőket is:

```
> evalm( B^(-3) );
```

$$\begin{bmatrix} -\frac{\%1\,\alpha}{-\alpha^2+\beta^2}-2\frac{\alpha\,\beta^2}{(-\alpha^2+\beta^2)^3} & \frac{\%1\,\beta}{-\alpha^2+\beta^2}+2\frac{\alpha^2\,\beta}{(-\alpha^2+\beta^2)^3} \\ \frac{\%1\,\beta}{-\alpha^2+\beta^2}+2\frac{\alpha^2\,\beta}{(-\alpha^2+\beta^2)^3} & -\frac{\%1\,\alpha}{-\alpha^2+\beta^2}-2\frac{\alpha\,\beta^2}{(-\alpha^2+\beta^2)^3} \end{bmatrix}$$

$$\%1 := \frac{\alpha^2}{(-\alpha^2+\beta^2)^2}+\frac{\beta^2}{(-\alpha^2+\beta^2)^2}$$

```
> evalm( " &* """ ):
```

Ügyeljünk arra, hogy a &* operátor után egy szóközt ki kell hagyni, másként a rendszer nem ismeri föl. A terjedelmes outputot nem írattuk ki, de megmutatjuk, hogy valóban egységmátrixot kaptunk.

```
> map( normal, " ); # the expected answer
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Az automatikus egyszerűsítés néha meglepő eredményekre vezet:

```
> evalm( A^0 );
```

$$1$$

```
> whattype(");
```

$$\mathit{integer}$$

Ügyelnünk kell továbbá az operátorok precedenciájára is, mivel a &* prioritása megegyezik a * (szorzás) és a / (osztás) prioritásával. Néha szükség lehet zárójelek használatára:

```
> evalm( A &* 1/A );

Error, (in evalm/amperstar)
\&* is reserved for matrix multiplication

> evalm( A &* (1/A) );
```

$$\&*()$$

A Maple az egységmátrixra a `&*()` jelölést használja. Az **lprint** alkalmazásakor jobban látszik, hogy miről is van szó valójában:

```
> lprint( A &* 1/A );

`\&*`(A,1)/A

> lprint( A &* (1/A) );

`\&*`(A,1/A)
```

18.4. Alapvető mátrixfüggvények

A **trace** és a **det** függvények a mátrix nyomát, illetve determinánsát számolják ki. Alkalmazzuk ezeket a következő Töplitz mátrixra:

```
> toeplitz( [1,2,3] );
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix}$$

```
> trace(");
```

$$3$$

```
> det("");
```

$$8$$

A mátrix sor- és oszloprangja, sor- és oszlopvektor-rendszerének, valamint magterének bázisa könnyen meghatározható:

```
> A := matrix( [ [1,0,0,1], [1,0,1,1], [0,0,1,0] ] );
```

$$A := \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> rank(A);
```

$$2$$

```
> rowspace(A), colspace(A);
```

$$\{[1, 0, 0, 1], [0, 0, 1, 0]\}, \{[1, 0, -1], [0, 1, 1]\}$$

```
> kernel(A);
```

$$\{[0, 1, 0, 0], [-1, 0, 0, 1]\}$$

Elvileg lehetőség van a karakterisztikus polinom, a sajátértékek és a sajátvektorok kiszámítására is, de az analitikus megoldás keresése nem megoldható karakterisztikus egyenlet esetében természetesen holtpontra juttathat bennünket.

```
> A := matrix( [ [-2,2,3], [3,7,-8], [10,-4,-3] ] );
```

$$A := \begin{bmatrix} -2 & 2 & 3 \\ 3 & 7 & -8 \\ 10 & -4 & -3 \end{bmatrix}$$

```
> cp := charpoly( A, lambda );
```

$$cp := \lambda^3 - 2\lambda^2 - 97\lambda + 282$$

A Cayley–Hamilton tétel azt állítja, hogy az A mátrixot a karakterisztikus polinomjába behelyettesítve zérusmátrixot kapunk. Ellenőrizzük ezt is.

```
> evalm( subs( lambda=A, cp ) );
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Az A sajátértékeinek meghatározásához meg kell oldani a karakterisztikus egyenletet.

```
> solve( cp );
```

$$3, -\frac{1}{2} + \frac{1}{2}\sqrt{377}, -\frac{1}{2} - \frac{1}{2}\sqrt{377}$$

A sajátértékeket egyből megkaphattuk volna a **eigenvalues** (vagy **eigenvals**) eljárással.

```
> eigenvalues( A );
```

$$3, -\frac{1}{2} + \frac{1}{2}\sqrt{377}, -\frac{1}{2} - \frac{1}{2}\sqrt{377}$$

A sajátvektorok az **eigenvectors** (vagy **eigenvects**) rutinnal számíthatók ki.

```
> eigenvectors( A );
```

$$\left[-\frac{1}{2} + \frac{1}{2}\sqrt{377}, 1, \left\{\left[-\frac{15}{38} + \frac{1}{38}\sqrt{377}, 1, \frac{15}{19} - \frac{1}{19}\sqrt{377}\right]\right\}\right],$$
$$\left[-\frac{1}{2} - \frac{1}{2}\sqrt{377}, 1, \left\{\left[-\frac{15}{38} - \frac{1}{38}\sqrt{377}, 1, \frac{15}{19} + \frac{1}{19}\sqrt{377}\right]\right\}\right],$$
$$\left[3, 1, \left\{\left[\frac{14}{13}, \frac{31}{26}, 1\right]\right\}\right]$$

Eredményül listákból álló sorozatot kaptunk. Minden lista egy sajátértéket, annak multiplicitását és sajátalterének egy bázisát tartalmazza. Alapértelmezés szerint a Maple gyökös kifejezésekkel írja föl a sajátértékeket. Ezt közvetlenül is kérhetjük a `radical` kulcsszó megadásával. Az `implicit` kulcsszó hatására implicit formában, a **RootOf** segítségével fölírt értékeket kapunk. Ezekből később az **allvalues** alkalmazásával kiszámíthatjuk az összes sajátértéket. A sajátvektorok is kérhetők ilyen **RootOf**-os alakban.

```
> eigenvalues( A, 'implicit' );
```

$$3,\ \mathrm{RootOf}(_Z^2+_Z-94)$$

```
> eigenvectors( A, 'implicit' );
```

$$[\mathrm{RootOf}(_Z^2+_Z-94),\ 1,\ \{\left[\frac{-1}{2},\ -\frac{1}{4}\mathrm{RootOf}(_Z^2+_Z-94)-2,\ 1\right]\}],$$
$$[3,\ 1,\ \{\left[\frac{14}{13},\ \frac{31}{26},\ 1\right]\}]$$

```
> map( allvalues, ["], 'dependent' );
```

$$[[-\frac{1}{2}+\frac{1}{2}\sqrt{377},\ 1,\ \{\left[\frac{-1}{2},\ -\frac{15}{8}-\frac{1}{8}\sqrt{377},\ 1\right]\}],$$
$$[-\frac{1}{2}-\frac{1}{2}\sqrt{377},\ 1,\ \{\left[\frac{-1}{2},\ -\frac{15}{8}+\frac{1}{8}\sqrt{377},\ 1\right]\}],\ [3,\ 1,\ \{\left[\frac{14}{13},\ \frac{31}{26},\ 1\right]\}]]$$

Eddig többnyire racionális számokból álló mátrixokkal dolgoztunk. Valójában a `linalg` csomag eljárásai a racionális együtthatós polinomok gyűrűje fölötti mátrixokra is alkalmazhatók. De sok eljárásnál még ennél is bővebb osztályokból választhatjuk az együtthatókat: sokszor komplex vagy algebrai számok, sőt algebrai függvények is megengedettek. Az alábbiakban egy Töplitz mátrixos példát mutatunk:

```
> A := toeplitz( [ sqrt(2), alpha, beta ] );
```

$$A := \begin{bmatrix} \sqrt{2} & \alpha & \beta \\ \alpha & \sqrt{2} & \alpha \\ \beta & \alpha & \sqrt{2} \end{bmatrix}$$

```
> factor( det(A), sqrt(2) );
```

$$(2\,\alpha^2-\beta\sqrt{2}-2)\,(-\sqrt{2}+\beta)$$

```
> charpoly( A, lambda );
```

$$\lambda^3-3\,\lambda^2\sqrt{2}+6\,\lambda-2\,\lambda\,\alpha^2-2\sqrt{2}+2\sqrt{2}\,\alpha^2-2\,\beta\,\alpha^2-\beta^2\,\lambda+\beta^2\sqrt{2}$$

```
> factor( ", sqrt(2) );
```

$$-(2\,\alpha^2-\lambda^2+2\,\lambda\sqrt{2}+\beta\,\lambda-2-\beta\sqrt{2})\,(-\sqrt{2}+\beta+\lambda)$$

```
> eigenvalues( A );
```

$$\sqrt{2}-\beta,\ \frac{1}{2}\beta+\sqrt{2}+\frac{1}{2}\sqrt{\beta^2+8\,\alpha^2},\ \frac{1}{2}\beta+\sqrt{2}-\frac{1}{2}\sqrt{\beta^2+8\,\alpha^2}$$

```
> eigenvectors( A );
```

$$\left[\frac{1}{2}\beta+\sqrt{2}+\frac{1}{2}\%1,\ 1,\ \left\{\left[1,\ -\frac{\frac{1}{2}\beta-\frac{1}{2}\%1}{\alpha},\ 1\right]\right\}\right],$$
$$\left[\frac{1}{2}\beta+\sqrt{2}-\frac{1}{2}\%1,\ 1,\ \left\{\left[1,\ -\frac{\frac{1}{2}\beta+\frac{1}{2}\%1}{\alpha},\ 1\right]\right\}\right],$$
$$[\sqrt{2}-\beta,\ 1,\ \{[-1,\ 0,\ 1]\}]$$
$$\%1 := \sqrt{\beta^2+8\alpha^2}$$

Némely függvények azonban nem engedik meg a lebegőpontos számok és a szimbolikus kifejezések egyidejű használatát.

```
> A := map( evalf, A );
```

$$A := \begin{bmatrix} 1.414213562 & \alpha & \beta \\ \alpha & 1.414213562 & \alpha \\ \beta & \alpha & 1.414213562 \end{bmatrix}$$

```
> eigenvectors( A );

Error, (in linalg/evalf)
matrix entries must all evaluate to float
```

A hibajelzés azt sugallja, hogy a Maple valamilyen numerikus sajátértékszámító rutinra szeretett volna áttérni, de ez nem sikerült, mivel a mátrix szimbolikus kifejezéseket is tartalmaz. Ilyen esetekben a következő trükkhöz folyamodhatunk: a lebegőpontos számokat racionálisakká konvertáljuk, elvégezzük a számításokat, végül numerikusan kiértékeljük az eredményt.

```
> Digits := 4: # low precision for display reasons
> A := map( convert, A, rational );
```

$$A := \begin{bmatrix} \frac{41}{29} & \alpha & \beta \\ \alpha & \frac{41}{29} & \alpha \\ \beta & \alpha & \frac{41}{29} \end{bmatrix}$$

```
> eigenvectors( A );
```

$$[\frac{41}{29}-\beta,\ 1,\ \{[-1,\ 0,\ 1]\}],$$
$$\left[\frac{41}{29}+\frac{1}{2}\beta+\frac{1}{2}\%1,\ 1,\ \left\{\left[\frac{1}{58}\,\frac{\frac{29}{2}\beta+\frac{29}{2}\%1}{\alpha},\ 1,\ \frac{1}{58}\,\frac{\frac{29}{2}\beta+\frac{29}{2}\%1}{\alpha}\right]\right\}\right],$$

$$\left[\frac{41}{29}+\frac{1}{2}\beta-\frac{1}{2}\%1,\ 1,\ \left\{\left[\frac{1}{58}\,\frac{\frac{29}{2}\beta-\frac{29}{2}\%1}{\alpha},\ 1,\ \frac{1}{58}\,\frac{\frac{29}{2}\beta-\frac{29}{2}\%1}{\alpha}\right]\right\}\right]$$

$$\%1 := \sqrt{\beta^2+8\,\alpha^2}$$

```
> map( evalf, ["] );
```

$$[[1.414-1.\,\beta,\ 1.,\ \{[-1.,\ 0,\ 1.]\}],\ [1.414+.5000\,\beta+.5000\,\%1,\ 1.,$$
$$\left\{\left[.01724\,\frac{14.50\,\beta+14.50\,\%1}{\alpha},\ 1.,\ .01724\,\frac{14.50\,\beta+14.50\,\%1}{\alpha}\right]\right\}],$$
$$[1.414+.5000\,\beta-.5000\,\%1,\ 1.,$$
$$\left\{\left[.01724\,\frac{14.50\,\beta-14.50\,\%1}{\alpha},\ 1.,\ .01724\,\frac{14.50\,\beta-14.50\,\%1}{\alpha}\right]\right\}]]$$
$$\%1 := \sqrt{\beta^2+8.\,\alpha^2}$$

A 18.2. táblázatban a `linalg` csomag használatakor rendelkezésünkre álló alapvető függvényeket tekintjük át.

Eljárás	A kiszámított objektum
adj, adjoint	a mátrix adjungáltja
charmat	a karakterisztikus mátrix
charpoly	a karakterisztikus polinom
cond	a mátrix kondíciószáma
det	a mátrix determinánsa
eigenvalues, eigenvals	a mátrix sajátértékei
eigenvectors, eigenvects	a mátrix sajátvektorai
hadamard	a Hadamard-féle korlát
htranspose	a mátrix Hermite-transzponáltja
inverse	a mátrix inverze
kernel	a mátrix magtere
minor	a mátrix minora
minpoly	a mátrix minimálpolinomja
permanent	a mátrix permanense
rank	a mátrix rangja
singularvals	a mátrix szinguláris értékei
trace	a mátrix nyoma
transpose	a mátrix transzponáltja

18.2. táblázat: A `linalg` csomag alapfüggvényei

18.5. A mátrixok struktúrájával kapcsolatos műveletek

A Maple sok olyan függvényt tartalmaz, amelyekkel a mátrixok struktúráján végezhetünk műveleteket. Ezeket soroljuk föl a 18.3. táblázatban.

Eljárás	Jelentése
addcol	a mátrix két oszlopának lineáris kombinációja
addrow	a mátrix két sorának lineáris kombinációja
augment, concat	mátrixok vízszintes összekapcsolása
blockmatrix	blokkos mátrix létrehozása
col	oszlop(ok) kivonása vektor(ok)ként a mátrixból
coldim	a mátrix oszloprangjának kiszámítása
copyinto	elemek átmásolása egyik mátrixból másikba
delcols	mátrix oszlopainak törlése
delrows	mátrix sorainak törlése
extend	a mátrix (méretének) megnövelése
mulcol	a mátrix oszlopának szorzása adott kifejezéssel
mulrow	a mátrix sorának szorzása adott kifejezéssel
row	sor(ok) kivonása vektor(ok)ként a mátrixból
rowdim	a mátrix sorrangjának kiszámítása
scalarmul	mátrix vagy vektor szorzása kifejezéssel
stack	mátrixok vagy vektorok függőleges összekapcsolása
submatrix	mátrixból részmátrix kiválasztása
subvector	mátrixból részvektor kiválasztása
swapcol	a mátrix két oszlopának fölszerélése
swaprow	a mátrix két sorának fölszerélése

18.3. táblázat: Strukturális műveletek

A `linalg` csomagban vannak olyan függvények is, melyek azt ellenőrzik, hogy az adott mátrix benne van-e valamely speciális osztályban, vagy meghatározott kapcsolatban áll-e egy másik mátrixszal.

Eljárás	Mit tesztel
definite	pozitív (vagy negatív) definit mátrix
equal	mátrixok egyenlősége
issimilar	mátrixok hasonlósága
iszero	zérusmátrix
orthog	ortogonális mátrix

18.4. táblázat: Mátrixok tesztfüggvényei

Elég lesz két példa. Kiindulásul vegyünk egy Jordan-féle blokkot és egy ugyanolyan dimenziós egységmátrixot:

```
> J := JordanBlock(2,4);
```

$$J := \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

```
> Id := band([1],4);
```

$$Id := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Egyesítsük vízszintesen a két mátrixot.

```
> augment( J, Id );
```

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A **gaussjord** eljárással fölső trianguláris alakra hozzuk a mátrixot:

```
> gaussjord(");
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{-1}{4} & \frac{1}{8} & \frac{-1}{16} \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{-1}{4} & \frac{1}{8} \\ 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{-1}{4} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

A jobboldali 4×4-es részmátrix az eredeti Jordan-blokk inverze.

```
> Jinv := submatrix( ", 1..4, 5..8 );
```

$$Jinv := \begin{bmatrix} \frac{1}{2} & \frac{-1}{4} & \frac{1}{8} & \frac{-1}{16} \\ 0 & \frac{1}{2} & \frac{-1}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{2} & \frac{-1}{4} \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

```
> equal( Jinv, inverse(J) );
```

true

A második példa a tesztfüggvények használatát demonstrálja. Tekintsük a következő két mátrixot:

```
> A := matrix( [
>        [ (sqrt(5)+sqrt(2)*sqrt(5-sqrt(5))+1)/4,
>        -sqrt(2)*sqrt(5-sqrt(5))/2 ],
>      [ sqrt(2)*sqrt(5-sqrt(5))/4,
>        (sqrt(5)-sqrt(2)*sqrt(5-sqrt(5))+1)/4 ] ] );
```

$$A := \begin{bmatrix} \frac{1}{4}\sqrt{5}+\frac{1}{4}\%1+\frac{1}{4} & -\frac{1}{2}\%1 \\ \frac{1}{4}\%1 & \frac{1}{4}\sqrt{5}-\frac{1}{4}\%1+\frac{1}{4} \end{bmatrix}$$

$$\%1 := \sqrt{2}\sqrt{5-\sqrt{5}}$$

```
> B := matrix( [
>      [ (sqrt(5)+1)/4, -sqrt(2)*sqrt(5-sqrt(5))/4],
>      [ sqrt(2)*sqrt(5-sqrt(5))/4, (sqrt(5)+1)/4 ] ] );
```

$$B := \begin{bmatrix} \frac{1}{4}\sqrt{5}+\frac{1}{4} & -\frac{1}{4}\sqrt{2}\sqrt{5-\sqrt{5}} \\ \frac{1}{4}\sqrt{2}\sqrt{5-\sqrt{5}} & \frac{1}{4}\sqrt{5}+\frac{1}{4} \end{bmatrix}$$

Belátjuk, hogy ezek a mátrixok hasonlóak, vagyis létezik olyan T nemszinguláris mátrix, amellyel $B = T\,A\,T^{-1}$.

```
> issimilar( A, B, 'T' );
```

true

```
> T := map( simplify, T );
```

$$T := \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix}$$

```
> equal( B,
>     map( simplify, evalm( T &* A &* inverse(T) ) ) );
```

true

B valójában ortogonális mátrix.

```
> orthog( B );
```

true

Ez a mátrix az óramutató járásával szembeni $\frac{\pi}{5}$ nagyságú szöggel való elforgatást reprezentál.

18.6. Vektorműveletek

A következő vektorműveletek állnak rendelkezésünkre. A vektoranalízisből vett példákat a 19.4. alfejezetben találhatja meg az Olvasó.

Eljárás	A kiszámított objektum
GramSchmidt	vektorok Gram–Schmidt ortogonalizálása
angle	vektorok szöge
basis	vektortér bázisa
colspace	mátrix oszlopterének bázisa
colspan	mátrix oszlopterét kifeszítő vektorok
crossprod	vektorok külső (vektoriális) szorzata
curl	vektor rotációja
diverge	vektorfüggvény divergenciája
dotprod, innerprod	vektorok belső szorzata
intbasis	vektorterek metszetének bázisa
laplacian	kifejezés Laplace-operátora
norm	vektor vagy mátrix normája
normalize	normalizált vektor
nullspace	a zéróaltér bázisa
potential	vektormező potenciálja
rowspace	mátrix sorterének bázisa
rowspan	mátrix sorterét kifeszítő vektorok
sumbasis	vektorterek összegének bázisa
vecpotent	vektor potenciálja
vectdim	vektor dimenziója

18.5. táblázat: Vektorműveletek

18.7. Mátrixok normálformái

A Maple-ben a mátrixok különböző normálformáival kapcsolatos lehetőségek is találhatók, melyeket a 18.6. táblázatban sorolunk föl. Néhány példa alapján jobban meg tudjuk ítélni ezek hasznosságát.

Eljárás	Normálforma
LUdecomp	*LU* dekompozíció
QRdecomp	*QR* dekompozíció
cholesky	Cholesky dekompozíció
exponential	mátrix exponenciális
ffgausselim	törtmentes Gauss-elimináció
frobenius, ratform	Frobenius (racionális kanonikus) normálforma
gausselim	Gauss-elimináció
gaussjord, rref	Gauss–Jordan forma
hermite	egyváltozós polinomok fölötti Hermite normálforma
ihermite	egészek fölötti Hermite normálforma
ismith	egészek fölötti Smith normálforma
jordan	Jordan normálforma
smith	egyváltozós polinomok fölötti Smith normálforma

18.6. táblázat: Mátrixok normálformái

A **Frobenius**, **Gausselim**, **Gaussjord**, **Hermite** és a **Smith** függvények a táblázatban fölsorolt kisbetűs függvények tétlen (inert) változatai. Ezeket véges testek vagy algebrai (függvény)testek fölötti számításoknál célszerű alkalmazni.

Első példánk a következő mátrix *LU* dekompozíciójának meghatározása:

$$A = \begin{pmatrix} 0 & -1 & -3 \\ 4 & 5 & 1 \\ 1 & 4 & -1 \end{pmatrix}.$$

Olyan L alsó trianguláris és U fölső trianguláris mátrixot keresünk, amelyekkel $A = PLU$ valamely P permutációs mátrixra.

```
> A := matrix( [[0,-1,-3], [4,5,1], [1,4,-1] ] );
```

$$A := \begin{bmatrix} 0 & -1 & -3 \\ 4 & 5 & 1 \\ 1 & 4 & -1 \end{bmatrix}$$

```
> LUdecomp( A, L='l', U='u', P='p' ):
> P = eval(p);
```

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> L = eval(l);
```

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & \frac{-11}{4} & 1 \end{bmatrix}$$

```
> U = eval(u);
```

$$U = \begin{bmatrix} 4 & 5 & 1 \\ 0 & -1 & -3 \\ 0 & 0 & \frac{-19}{2} \end{bmatrix}$$

```
> equal( evalm( p &* l &* u ), A );
```

true

Az A pozitív definit mátrix Cholesky-dekompozíciója olyan L alsó trianguláris mátrix, amelyre $A = LL^t$. A Maple-lel kiszámítattjuk a következő mátrix dekompozícióját.

```
> A := matrix( [ [5,-2,2], [-2,5,1], [2,1,2] ] );
```

$$A := \begin{bmatrix} 5 & -2 & 2 \\ -2 & 5 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

```
> L := cholesky( A );
```

$$L := \begin{bmatrix} \sqrt{5} & 0 & 0 \\ -\frac{2}{5}\sqrt{5} & \frac{1}{5}\sqrt{105} & 0 \\ \frac{2}{5}\sqrt{5} & \frac{3}{35}\sqrt{105} & \frac{1}{7}\sqrt{21} \end{bmatrix}$$

```
> equal( L &* transpose(L), A );
```

true

Legyen $A(x)$ testfölötti polinomokból álló mátrix. Ekkor léteznek ezen test fölött olyan P és Q mátrixok, amelyekkel

$$A = P\,\mathrm{diag}(a_1(x), a_2(x), \ldots, a_k(x), 0, 0, \ldots, 0)\,Q,$$

ahol az a_i polinom osztója a_{i+1}-nek. Az a_i-ket invariáns tényezőknek nevezzük; ezek a test nemnulla elemeivel való szorzás erejéig egyértelműek, és a k számot is egyértelműen meghatározza az $A(x)$ mátrix. A diagonális mátrixot az $A(x)$ Smith normálformájának hívjuk. Tekintsük a $\mathbb{Q}[x]$ fölötti következő mátrixot:

```
> A := matrix( [ [1,0,-1,1], [-1,x+1,-x,-x-2],
>   [-1,x+1, 2*x^3+2*x^2-3*x-2, 2*x^3+2*x^2-3*x-4],
>   [0,-x-1,2*x^3+2*x^2-x-1,2*x^3+2*x^2-x-1] ] );
```

$$A := \begin{bmatrix} 1 & 0 & -1 & 1 \\ -1 & x+1 & -x & -x-2 \\ -1 & x+1 & 2x^3+2x^2-3x-2 & 2x^3+2x^2-3x-4 \\ 0 & -x-1 & 2x^3+2x^2-x-1 & 2x^3+2x^2-x-1 \end{bmatrix}$$

A Smith normálformát a következő Maple paranccsal kapjuk:

```
> S := smith( A, x, 'U', 'V' );
```

$$S := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x+1 & 0 & 0 \\ 0 & 0 & x^3+x^2-x-1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ellenőrizzük az eredményt.

```
> P := inverse( U );
```

$$P := \begin{bmatrix} 1 & -3+4x^2 & 4 & \frac{-1}{2} \\ -1 & 2-2x^2 & -2 & \frac{1}{2} \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

```
> Q := inverse( V );
```

$$Q := \begin{bmatrix} 1 & -x-1 & -2x^3-2x^2+3x+2 & -2x^3-2x^2+3x+4 \\ 0 & 1 & 1-2x^2 & 1-2x^2 \\ 0 & -1 & 2x^2 & 2x^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> equal( map( expand, evalm( P &* S &* Q ) ), A );
```

true

Most tekintsük $A(x)$-et $\mathbb{F}_2[x]$ fölötti mátrixnak, vagyis számoljunk modulo 2.

```
> A := map( `mod`, A, 2 );
```

$$A := \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & x+1 & x & x \\ 1 & x+1 & x & x \\ 0 & x+1 & x+1 & x+1 \end{bmatrix}$$

```
> Smith( A, x ) mod 2;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x+1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figyeljük meg, hogy ez nem ugyanaz, mintha először kiszámolnánk a $\mathbb{Q}[x]$ fölötti normálformát, és erre alkalmaznánk moduláris aritmetikát.

```
> map( 'mod', smith( A, x ), 2 );
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x+1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ugyanez a jelenség előfordul a többi normálformával kapcsolatban is. Határozzuk meg ezeket az utóbbi A mátrixra.

```
> A = eval(A);
```

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & x+1 & x & x \\ 1 & x+1 & x & x \\ 0 & x+1 & x+1 & x+1 \end{bmatrix}$$

```
> F := frobenius( A, 'P' ); # over Q(x)
```

$$F := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2+2x \\ 0 & 1 & 0 & -4x-2 \\ 0 & 0 & 1 & 3x+3 \end{bmatrix}$$

```
> equal( map( normal, evalm( P &* F &* P^(-1) ) ), A );
```

true

```
> F := Frobenius( A, 'P' ) mod 2; # over F_2[x]
```

$$F := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & x+1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ellenőrizzük az eredményt:

```
> map( Normal, evalm( P &* F &* (Inverse(P) mod 2) ) ):
> equal( map('mod',",2), A );
```

true

```
> H := map( expand, hermite( A, x, 'T' ) ); # over Q[x]
```

$$H := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x+1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> equal( map( expand, evalm( T &* A ) ), H );
```

true

```
> H := Hermite( A, x, 'T' ) mod 2; # over F_2[x]
```

$$H := \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & x+1 & x+1 & x+1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> equal( map( `mod`, evalm( T &* A ), 2 ), H );
```

true

```
> gausselim(A); # Gaussian elimination over Z(x)
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & x+1 & x-1 & x-1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> ffgausselim(A); # fraction-free Gaussian elimination
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & x+1 & x-1 & x-1 \\ 0 & 0 & 2+2x & 2+2x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> Gausselim(A) mod 2; # Gaussian elimination over F_2[x]
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & x+1 & x+1 & x+1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> gaussjord(A); # Gauss-Jordan form over Q(x)
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> Gaussjord(A) mod 2; # Gauss-Jordan form over F_2(x)
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

18.8. Gyakorlatok

1. Tekintsük a következő mátrixokat

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{pmatrix}, \qquad B = \begin{pmatrix} -3 & 2 \\ 0 & 1 \\ 7 & 4 \end{pmatrix}$$

és számítsuk ki az alábbi kifejezéseket

(a) A^{-1},

(b) AA^t,

(c) B^tAB,

(d) $(2A + BB^t)A^t$.

2. Számítsuk ki a $[\cos x, \sin x, e^x]$ függvényekből álló lista Wronski mátrixát és a mátrix determinánsát. A $[\cosh x, \sinh x, e^x]$ listával is végezzük el ugyanezeket a számításokat.

3. Hozzunk létre véletlenszerűen választott egész együtthatós polinomokból álló 5×5-ös mátrixot. A polinomok legföljebb negyedfokúak legyenek, és három (nemnulla) tagot tartalmazzanak. Számítsuk ki a karakterisztikus polinomot, és ellenőrizzük a Cayley–Hamilton tétel érvényességét a fő változója szerint gyűjtött alakra hozott karakterisztikus polinomba behelyettesítve a mátrixot.

Számoljuk ki a karakterisztikus polinom Horner-elrendezéses alakját is, és ellenőrizzük a Cayley–Hamilton tétel érvényességét a polinom Horner-elrendezéses alakjába behelyettesítve a mátrixot. Hasonlítsuk össze a műveleti időket.

4. Tekintsük az

$$A = \begin{pmatrix} -4 & -7 & 0 \\ 0 & 4 & 2 \\ -5 & -7 & 1 \end{pmatrix} \text{ és a } B = \begin{pmatrix} -2 & -1 & -2 \\ 2 & 2 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

mátrixokat. Mutassuk meg, hogy A és B hasonlóak, és számítsuk ki a megfelelő transzformációs mátrixot.

5. Legyen $a, b \in \mathbb{R}$, $0 \leq a \leq 1$, $b^2 = 2a(1-a)$ és

$$A = \begin{pmatrix} a & a-1 & b \\ a-1 & a & b \\ -b & -b & 2a-1 \end{pmatrix}.$$

(a) Ellenőrizzük a Maple-lel, hogy A 1 determinánsú ortogonális mátrix.

(b) (a)-ból következik, hogy A az $\mathbb{R}^3$ standard bázisával leírható elforgatásnak felel meg. Határozzuk meg az elforgatás szögét.

6. Legyen $a, b \in \mathbb{R}$ és

$$A = \begin{pmatrix} 0 & a & 1 & 0 & b \\ 1 & 0 & 0 & b & 0 \\ 0 & 1 & b & 0 & 1 \\ b & 0 & 0 & 1 & 0 \\ 0 & b & 1 & 0 & b \end{pmatrix}.$$

(a) Az a és b mely értékeire szinguláris az A mátrix?

(b) Határozzuk meg A inverzét (azokra az a és b értékekre, amelyekre A invertálható).

7. Számítsuk ki a következő determinánsokat:

(a) $\det \begin{pmatrix} x^2+1 & x & 0 & 0 \\ x & x^2+1 & x & 0 \\ 0 & x & x^2+1 & x \\ 0 & 0 & x & x^2+1 \end{pmatrix}$,

(b) $\det \begin{pmatrix} x^2+1 & x & 0 & 0 & 0 \\ x & x^2+1 & x & 0 & 0 \\ 0 & x & x^2+1 & x & 0 \\ 0 & 0 & x & x^2+1 & x \\ 0 & 0 & 0 & x & x^2+1 \end{pmatrix}$.

(c) Az (a) és (b) feladatok eredményét látva van-e valami ötletünk arra, hogy az ilyen alakú általános mátrixnak mi a determinánsa? Ha igen, akkor ellenőrizzük sejtésünket a 8×8-as esetre. Ha nem, akkor számítsuk ki a 6×6-os és a 7×7-es determinánsokat is, hátha ezek alapján észreveszünk valami törvényszerűséget.

8. Minden n természetes számra definiáljuk a következő A_n $n \times n$-es mátrixot:

$$A_n(i,j) = \begin{cases} 0, & \text{ha } i = j, \\ 1, & \text{ha } i \neq j. \end{cases}$$

Hajtsuk végre a következő számításokat az $n = 3, 4$ és 5 értékekre.

(a) Számítsuk ki A_n determinánsát.

(b) Számítsuk ki A_n karakterisztikus polinomját.

(c) Határozzuk meg A_n összes sajátértékét és az egyes sajátértékekhez tartozó sajátalterek egy-egy bázisát.

9. Minden n természetes számra definiáljuk a következő A_n $n \times n$-es mátrixot:

$$A_n(i,j) = \gcd(i,j).$$

(a) Számítsuk ki az A_n mátrix determinánsát $n = 1, 2, \ldots, 15$-re.

(b) Próbáljunk zárt formulát találni az általános esetre (inkább matematikusok számára ajánlott).

10. Határozzuk meg az [50]-ből vett következő mátrix LU dekompozícióját:

$$A = \begin{pmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix}.$$

11. Legyen A a kételemű test fölötti következő mátrix:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Határozzunk meg olyan $\mathbb{F}_2$ fölötti T transzformációs mátrixot, amellyel $T^{-1}AT$

$$\begin{pmatrix} B & 0 \\ 0 & C \end{pmatrix}$$

alakú, ahol B az $\mathbb{F}_2$ fölötti 2×2-es, C pedig az $\mathbb{F}_2$ fölötti 3×3-as mátrix.

[illegible]

8. [illegible]

$$A_n(i,j) = \begin{cases} 0, & \text{ha } i = j, \\ 1, & \text{ha } i \neq j. \end{cases}$$

[illegible]

(a) [illegible]

(b) [illegible]

(c) Határozzuk meg, [illegible]

9. Tekintsük [illegible] a következő A_n [illegible] mátrixot:

$$A_n(i,j) = \text{lnko}(i,j).$$

(a) Számítsuk ki az A_n mátrix determinánsát [illegible].

(b) Próbáljunk [illegible] találni az [illegible] (… számára ajánlott).

10. Határozzuk meg az [illegible] mátrix [illegible] determinánsát:

[illegible]

11. Legyen [illegible]

[illegible]

[illegible]

[illegible]

19.

A lineáris algebra alkalmazásai

A fejezet a mátrixokkal kapcsolatos számításokat a következő öt gyakorlati példával illusztrálja:

- a Stanford manipulator kinematikája,
- az emberi test kadmium-fölvételének háromszakaszos modellje,
- a molekula-pályák Hückel-féle elmélete,
- vektoranalízis és
- a Moore–Penrose általánosított inverz.

Minden szekcióban föltesszük, hogy a `linalg` csomagot előzetesen betöltöttük.

19.1. A Stanford manipulátor kinematikája

Az alábbi mátrixok az úgynevezett Denavit–Hartenberg mátrixok, melyeket a robotmanipulátorok kinematikájának vizsgálatánál alkalmaznak. Tulajdonképpen a Stanford manipulátort definiáló $A_1, A_2, \ldots, A_6$ mátrixokat fogjuk használni (lásd [98]). A további Maple szekciókban a $c1 = \cos\theta_1, c2 = \cos\theta_2, \ldots$ továbbá az $s1 = \sin\theta_1, s2 = \sin\theta_2, \ldots$ rövidítésekkel dolgozunk.

```
> alias( seq( c[i] = cos(theta[i]), i=1..6 ),
>   seq( s[i] = sin(theta[i]), i=1..6 ) ):
> M := (a,alpha,d,theta) -> matrix( 4, 4, [cos(theta),
>   -sin(theta)*cos(alpha), sin(theta)*sin(alpha),
```

```
>     a*cos(theta), sin(theta), cos(theta)*cos(alpha),
>     -cos(theta)*sin(alpha), a*sin(theta), 0, sin(alpha),
>     cos(alpha), d, 0, 0, 0, 1 ] ):
>   M(a,alpha,d,theta);
```

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & a\cos(\theta) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>   # link length
>   a := vector([0$6]):
>   # link twist
>   alpha := vector([-Pi/2,Pi/2,0,-Pi/2,Pi/2,0]):
>   # offset distance
>   d := vector([0$6]):
>   d[2] := evaln(d[2]):  d[3] := evaln(d[3]):
>   # joint angle
>   theta := vector(6):  theta[3] := 0:
>   for i to 6 do
>     A[i] := M( a[i], alpha[i], d[i], theta[i] )
>   od;
```

$$A_1 := \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 := \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 := \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 := \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 := \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A manipulátor csúcsának helyzetét és irányát a Denavit–Hartenberg paraméterek határozzák meg: az $A_1, A_2, \ldots, A_6$ mátrixot szorzatát kell vennünk.

```
> Tip := evalm( `&*`( seq( A[i], i=1..6 ) ) ):
```

Nézzük meg a bal fölső sarokban álló elemet:

```
> collect( Tip[1,1], [c[1],c[2],s[1]] );
```

$$((c_4\, c_5\, c_6 - s_4\, s_6)\, c_2 - s_2\, s_5\, c_6)\, c_1 + (-s_4\, c_5\, c_6 - c_4\, s_6)\, s_1$$

Ez megegyezik a [155]-ben található 2.55. képlettel. Ilyen mátrixoknál a kézi számoláshoz jóval több idő kellene, és a hibákat is nehezebb lenne elkerülni. A számítógépes algebrai rendszer gond nélkül ledarálta[1] a szimbolikus számításokat.

A [81] irodalomban arra találhatunk példát, hogyan segíthet a számítógépes algebra az inverz kinematikai feladat megoldásában. (Itt a link paraméterek olyan értékeinek meghatározása a feladat, amelyekkel a robotkar adott pozícióba és irányításba vihető át.)

Ebben a részben az eredeti kinematikai problémát, még pontosabban a Stanford manipulátor csúcsának sebességét vizsgáljuk. Ha a robotkar fizikai paramétereit (a d_2 és a d_3 értékeket) rögzítettük, a robotkar csúcsa a $\theta_1, \theta_2, \theta_4, \theta_5$ és a θ_6 függvényeként adható meg. A csúcs v transzlációs és ω forgássebességét a $\theta_1', \theta_2', \theta_4', \theta_5'$ és a θ_6' függvényében írjuk föl. Először szétválasztjuk a csúcs pozícióját és irányítását megadó mátrix T transzlációs és R rotációs részét. Ezt könnyen megtehetjük a **subvector** és a **submatrix** eljárások segítségével.

```
> T := subvector( Tip, 1..3, 4 );
```

$$T := [\, c_1\, s_2\, d_3 - s_1\, d_2,\; s_1\, s_2\, d_3 + c_1\, d_2,\; c_2\, d_3 \,]$$

[1] Angolul a „number cruncher”, „symbolic cruncher” kifejezéseket szokták ilyen esetekben használni. (*A Fordító megjegyzése.*)

```
> R := submatrix( Tip, 1..3, 1..3 ):
```

A transzlációs- és a forgássebességre vonatkozó információkat a

$$J = \begin{pmatrix} J_1^v & J_2^v & \dots & J_5^v \\ J_1^\omega & J_2^\omega & \dots & J_5^\omega \end{pmatrix}$$

Jacobi mátrix tartalmazza, ahol $J = (\, J_1^v \; J_2^v \; \dots \; J_5^v \,)$ annak a vektorfüggvénynek a Jacobi mátrixa, amely a $\theta_1, \theta_2, \theta_4, \theta_5$ és a θ_6 kinematikai paramétereket képezi le a T transzlációs részmátrixba. A J_i^ω-k

$$J_i^\omega = \begin{pmatrix} \omega_{x_i} \\ \omega_{y_i} \\ \omega_{z_i} \end{pmatrix}$$

alakú 3×1-es vektorok, amelyek komponenseit a

$$\frac{\partial R}{\partial \theta_i} \cdot R^T = \begin{pmatrix} 0 & -\omega_{z_i} & \omega_{y_i} \\ \omega_{z_i} & 0 & -\omega_{x_i} \\ -\omega_{y_i} & \omega_{x_i} & 0 \end{pmatrix}$$

összefüggésekből kapjuk. Először tekintsük a transzlációs részt. A `linalg` csomag tartalmazza a vektorértékű függvények Jacobi mátrixát kiszámoló **jacobian** eljárást.

```
> Jv := jacobian( T,
>    [theta[1],theta[2],theta[4],theta[5],theta[6]] );
```

$$Jv := \begin{bmatrix} -s_1\, s_2\, d_3 - c_1\, d_2 & c_1\, c_2\, d_3 & 0 & 0 & 0 \\ c_1\, s_2\, d_3 - s_1\, d_2 & s_1\, c_2\, d_3 & 0 & 0 & 0 \\ 0 & -s_2\, d_3 & 0 & 0 & 0 \end{bmatrix}$$

Ezután a csúcs forgássebességét tekintjük. A $\dfrac{\partial R}{\partial \theta_i}$ deriváltak meghatározását a **diff** eljárásnak az összes mátrixelemre való leképezésével végezhetjük.

```
> R1 := map( diff, R, theta[1] ):
> R2 := map( diff, R, theta[2] ):
> R3 := map( diff, R, theta[4] ):
> R4 := map( diff, R, theta[5] ):
> R5 := map( diff, R, theta[6] ):
```

Mátrix transzponáltját (egyáltalán nem meglepő módon) a **transpose** eljárással képezhetjük.

```
> Rtranspose := transpose( R ):
```

Most már minden rendelkezésünkre áll a J_1^ω, J_2^ω, ..., J_5^ω mátrixok kiszámításához. Példaként a J_1^ω-ra kapott eredményt mutatjuk meg:

```
> evalm( R1 &* Rtranspose ): omega := map( simplify, " );
```

$$\omega := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
> Jomega[1] := vector([omega[3,2],-omega[3,1],omega[2,1]]);
```

$$Jomega_1 := [0, 0, 1]$$

Az összes többi vektort megkaphatjuk egyetlen ciklusutasítással. Befejezésül a **concat** vagy az **augment** eljárással rakhatjuk össze a kívánt J részmátrixot:

```
> for i from 2 to 5 do
>   evalm( R.i &* Rtranspose ):
>   omega := map( simplify, " ):
>   Jomega[i] := vector( [ omega[3,2], -omega[3,1],
>     omega[2,1] ] ):
>   print( Jomega[i]=eval(Jomega[i]) )
> od:
```

$$Jomega_2 = [-s_1, c_1, 0]$$

$$Jomega_3 = [c_1 s_2, s_1 s_2, c_2]$$

$$Jomega_4 = [-c_1 c_2 s_4 - s_1 c_4, -s_1 c_2 s_4 + c_1 c_4, s_4 s_2]$$

$$Jomega_5 = \Big[- s_1 s_5 s_4 + c_1 s_2 c_5 + c_1 c_4 c_2 s_5,$$
$$s_1 c_4 c_2 s_5 + c_1 s_5 s_4 + s_1 s_2 c_5, -s_2 c_4 s_5 + c_2 c_5\Big]$$

```
> Jomega := concat( seq( eval(Jomega[i]), i=1..5 ) ):
```

A `linalg` csomag **stack** eljárásával helyezhetjük el egyetlen mátrixba egymás alá a J transzlációs és rotációs részét:

```
> J := stack( Jv, Jomega );
```

$$J := \begin{bmatrix} -s_1 s_2 d_3 - c_1 d_2 & c_1 c_2 d_3 & 0 & 0 & 0 \\ c_1 s_2 d_3 - s_1 d_2 & s_1 c_2 d_3 & 0 & 0 & 0 \\ 0 & - s_2 d_3 & 0 & 0 & 0 \\ 0 & - s_1 & c_1 s_2 & - c_1 c_2 s_4 - s_1 c_4 & - s_1 s_5 s_4 + c_1 s_2 c_5 + c_1 c_4 c_2 s_5 \\ 0 & c_1 & s_1 s_2 & - s_1 c_2 s_4 + c_1 c_4 & s_1 c_4 c_2 s_5 + c_1 s_5 s_4 + s_1 s_2 c_5 \\ 1 & 0 & c_2 & s_4 s_2 & - s_2 c_4 s_5 + c_2 c_5 \end{bmatrix}$$

Tekintsük a Jacobi mátrix transzlációs részét:

```
> print( Jv = eval(Jv) );
```

$$Jv = \begin{bmatrix} -s_1 s_2 d_3 - c_1 d_2 & c_1 c_2 d_3 & 0 & 0 & 0 \\ c_1 s_2 d_3 - s_1 d_2 & s_1 c_2 d_3 & 0 & 0 & 0 \\ 0 & -s_2 d_3 & 0 & 0 & 0 \end{bmatrix}$$

A mátrix maximális rangja kettő. Ha valamely kinematikai paraméter-értékekre a rang ennél kisebb, azt mondjuk, hogy a manipulátor szinguláris állapotban van. A szinguláris állapotok a robotkar tiltott konfigurációi. Próbáljunk ilyen paraméter-értékeket keresni. Mi történik, ha $s_2 = 0$ (vagyis $\theta_2 \in \{0, \pi\}$)?

```
> subs( s[2]=0, eval(Jv) );
```

$$\begin{bmatrix} -c_1\,d_2 & c_1\,c_2\,d_3 & 0 & 0 & 0 \\ -s_1\,d_2 & s_1\,c_2\,d_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> submatrix( ", 1..2, 1..2 );
```

$$\begin{bmatrix} -c_1\,d_2 & c_1\,c_2\,d_3 \\ -s_1\,d_2 & s_1\,c_2\,d_3 \end{bmatrix}$$

Ennek a részmátrixnak a determinánsát a **det**-tel számíthatjuk ki:

```
> det(");
```

$$0$$

A rang tehát kisebb kettőnél, és a manipulátor szinguláris állapotban van. Ha $s_2 \neq 0$, elemi sorműveletekkel vizsgálhatjuk a soralteret. Például az **addrow** eljárással a harmadik sor $c_2 c_1 s_2$-szeresét hozzá tudjuk adni az első sorhoz:

```
> addrow( Jv, 3, 1, c[2]*c[1]/s[2] );
```

$$\begin{bmatrix} -s_1\,s_2\,d_3 - c_1\,d_2 & 0 & 0 & 0 & 0 \\ c_1\,s_2\,d_3 - s_1\,d_2 & s_1\,c_2\,d_3 & 0 & 0 & 0 \\ 0 & -s_2\,d_3 & 0 & 0 & 0 \end{bmatrix}$$

```
> addrow( ", 3, 2, s[1]*c[2]/s[2] );
```

$$\begin{bmatrix} -s_1\,s_2\,d_3 - c_1\,d_2 & 0 & 0 & 0 & 0 \\ c_1\,s_2\,d_3 - s_1\,d_2 & 0 & 0 & 0 & 0 \\ 0 & -s_2\,d_3 & 0 & 0 & 0 \end{bmatrix}$$

A rang akkor és csak akkor kisebb kettőnél, ha az első oszlop zérus:

```
> { "[1,1]=0, "[2,1]=0 };
```

$$\{-s_1\,s_2\,d_3 - c_1\,d_2 = 0,\ c_1\,s_2\,d_3 - s_1\,d_2 = 0\}$$

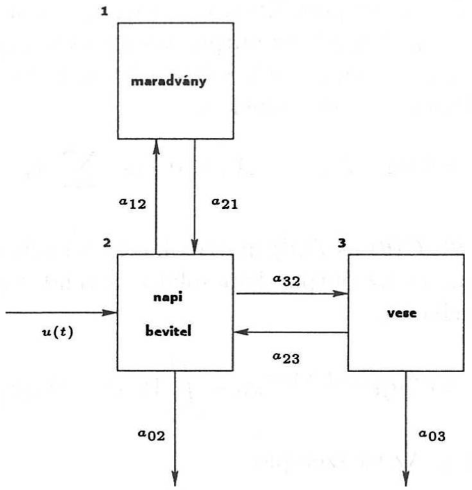
```
> solve( ", {s[1],c[1]} );
```

$$\{s_1 = 0,\ c_1 = 0\}$$

Ez csak akkor lehetne igaz, ha mind θ_1, mind θ_2 nulla, de ez lehetetlen a definíciók miatt. Tehát a manipulátornak nem találtunk további szinguláris állapotait.

19.2. A kadmium-fölvétel háromszakaszos modellje

A következő példa a lineáris rendszerek elméletéből származó esettanulmány. A [25] és a [99] irodalmakban olyan idő-invariáns folytonos idejű szakaszos modellek strukturális azonosíthatóságát vizsgálták, amelyek a kadmium áramlását írják le az emberi testben. Mi csak a 19.1. ábrának megfelelő háromszakaszos modellt tekintjük.



19.1. ábra: A kadmium fölhalmozódása az emberi szervezetben

A megfelelő matematikai modell a következő differenciálegyenlet-rendszer:

$$\begin{aligned} \frac{dx(t)}{dt} &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t), \end{aligned}$$

ahol

$$A = \begin{pmatrix} -a_{21} & a_{12} & 0 \\ a_{21} & -(a_{02}+a_{12}+a_{32}) & a_{23} \\ 0 & a_{32} & -(a_{03}+a_{23}) \end{pmatrix}$$

és

$$B = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 0 & c_3 \\ 0 & a_{02} & a_{03} \end{pmatrix}.$$

Föltesszük, hogy minden paraméter nemnegatív. A kadmium koncentrációját két helyen mérjük. A vesében elvégzett mérést az

$$y_1(t) = c_3 x_3(t),$$

a vizeletben mért értékeket az

$$y_2(t) = a_{02}x_2(t) + a_{03}x_3(t)$$

összefüggéssel írjuk le. Föltesszük, hogy c_3 ismert pozitív értékű paraméter.

A θ paramétertől függő $M(\theta)$ idő-invariáns folytonos idejű n-szakaszos lineáris modell általános alakja

$$M(\theta):\left\{\begin{array}{rcll} \dot{x}(t) & = & A(\theta)\cdot x(t)+B(\theta)\cdot u(t), & x(t_0)=x_0, \\ \dot{y}(t) & = & C(\theta)\cdot x(t)+D(\theta)\cdot u(t), & \end{array}\right.$$

ahol $A(\theta)$ $n\times n$-es, $B(\theta)$ $n\times m$-es, $C(\theta)$ $k\times n$-es, $D(\theta)$ $k\times m$-es mátrix, $x=(x_1,x_2,\dots,x_n)^T\in\mathbb{R}_+^n$ az állapotvektor, $u=(u_1,u_2,\dots,u_m)^T\in\mathbb{R}_+^m$ az input vektor, $y=(y_1,y_2,\dots,y_k)^T\in\mathbb{R}_+^k$ az output vektor és végül $\theta=(\theta_1,\theta_2,\dots,\theta_r)$ a paraméter-vektor. ($\mathbb{R}_+$ a nemnegatív valós számok halmazát jelöli.) Az $A(\theta)$ rendszermátrix együtthatóira teljesülnek az

$$A_{ij}\geq 0 \text{ ha } i\neq j, \qquad A_{ii}\leq 0 \quad \text{és} \quad \sum_i A_{ji}\leq 0$$

összefüggések. A $B(\theta)$, $C(\theta)$ és $D(\theta)$ mátrixok összes együtthatója nemnegatív valós szám. Az input és az output kapcsolatát a rendszer úgynevezett *külső viselkedése* írja le, melyet az

$$y(t)=C(\theta)e^{(t-t_0)A(\theta)}x_0+\int_{t_0}^{t}W_\theta(t-\tau)u(\tau)\,d\tau$$

függvény határoz meg. Az itt szereplő

$$W_\theta(t)=C(\theta)e^{tA(\theta)}B(\theta)+D(\theta)\delta(t)$$

függvényt a rendszer *impulzus válaszfüggvényének* hívják. Ezt a függvényt egyértelműen meghatározzák az alábbi, úgynevezett *Markov-féle paramétermátrixok*:

$$\begin{array}{rcl} M_0(\theta) & = & D \\ M_j(\theta) & = & \dfrac{d^{k-1}}{dt^{k-1}}W_\theta(t)\Big|_{x=0} \\ & = & C(\theta)A(\theta)^{k-1}B(\theta), \quad \text{ahol } k=1,2,\dots \end{array}$$

A továbbiakban föltesszük, hogy $D(\theta)=0$.

Az elméleti (a priori) azonosíthatóság tanulmányozásához három rendszerelméleti fogalom kacsolódik: a *vezérelhetőség*, a *megfigyelhetőség* és a *strukturális azonosíthatóság*. Rövid bevezetést és a számítógépes algebra alkalmazására vonatkozó megjegyzéseket a [126, 157] irodalmakban találhat az Olvasó. A következő kritériumok érvényesek:

A Kálmán-féle vezérelhetőségi kritérium. *A lineáris rendszer akkor és csak akkor vezérelhető, ha az*

$$M_C=(B|AB|A^2B|\cdots|A^{n-rb}B)$$

vezérelhetőségi mátrix maximális rangja n (Itt $rb=\operatorname{rang}(B)$*.)*

A Kálmán-féle megfigyelhetőségi kritérium. *A lineáris rendszer akkor és csak akkor megfigyelhető, ha az*

$$M_O = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-rc} \end{pmatrix}$$

megfigyelhetőségi mátrix maximális rangja n (Itt $rc = \text{rang}(C)$.*)*

Megjegyzés: a fönti kritériumokban vehetjük mindig az $rb = rc = 1$ értékeket is, például ha nem ismerjük vagy nem akarjuk kiszámítani C rangját.

Ezek után a paraméterek azonosításának alapfeladata röviden így fogalmazható meg:

> „Elvileg lehetséges-e a rendszer inputjának és outputjának ismeretében egyértelműen meghatározni a paramétereket?”

Emlékeztetünk arra, hogy algebrai halmazon polinomok halmazának közös zérushelyeiként megadható halmazt értünk. A strukturális azonosíthatósággal kapcsolatban a következő definíciókat fogjuk fölhasználni:

Az $M(\theta)$ modell a θ helyen *strukturálisan lokálisan azonosítható*, ha megadható θ-nak olyan Ω nyitott környezete, amely nem tartalmaz olyan θ-tól különböző θ'-t, amely minden (t_0, x_0, u) azonosítási kísérletre a θ-val megegyező külső viselkedést eredményezne.

Az $M(\theta)$ modell *strukturálisan lokálisan azonosítható*, ha legföljebb egy algebrai halmaz elemeit kivéve minden más θ helyen *strukturálisan lokálisan azonosítható.*

Az $M(\theta)$ modell a θ helyen *strukturálisan globálisan azonosítható*, ha nincs olyan θ-tól különböző θ', amely minden (t_0, x_0, u) azonosítási kísérletre a θ-val megegyező külső viselkedést eredményezne.

Az $M(\theta)$ modell *strukturálisan globálisan azonosítható*, ha legföljebb egy algebrai halmaz elemeit kivéve minden más θ helyen *strukturálisan globálisan azonosítható.*

A kadmium áramlását az emberi testben leíró modellünkkel kapcsolatban föltettük, hogy a rendszer stabil, és az időhorizont a rendszer dinamikájához képest viszonylag tágas. Ezért a kezdeti föltétel lényegében elhagyható. Ekkor az $u(t)$ input és az $y(t)$ output kapcsolatát az $y = W_\theta * u$ konvolúció határozza meg. Az alábbiakban a strukturális azonosíthatóság meghatározásának három módszerét említjük.

A Markov-féle paramétermátrixos módszer. *Konstruáljuk meg a következő mátrixot az* $M_1(\theta)$, $M_2(\theta), \ldots, M_{2n}(\theta)$ *paramétermátrixok segítségével:*

$$M_\theta = \begin{pmatrix} C(\theta)B(\theta) \\ C(\theta)A(\theta)B(\theta) \\ C(\theta)A(\theta^2)B(\theta) \\ \vdots \\ C(\theta)A(\theta^{2n-1})B(\theta) \end{pmatrix}$$

A lineáris rendszer akkor és csak akkor strukturálisan globálisan azonosítható, ha legföljebb egy algebrai halmaz elemeit kivéve minden más θ-ra $M_\theta = M'_\theta$-ből $\theta = \theta'$ következik. A lineáris rendszer akkor és csak akkor strukturálisan lokálisan azonosítható, ha a

$$\begin{pmatrix} \frac{\partial M_1}{\partial \theta_1} & \cdots & \frac{\partial M_1}{\partial \theta_r} \\ \vdots & & \vdots \\ \frac{\partial M_{2n}}{\partial \theta_1} & \cdots & \frac{\partial M_{2n}}{\partial \theta_r} \end{pmatrix}$$

$2nkm \times r$-es Jacobi mátrix legföljebb egy algebrai halmaz elemeit kivéve minden más θ-ra maximális rangú.

Az átvitel-függvényes módszer. *Definiáljuk a $H_\theta(s)$ átvitelfüggvényt a következő módon:*

$$H_\theta(s) = C(\theta)[sI - A(\theta)]^{-1}B(\theta).$$

A lineáris rendszer akkor és csak akkor strukturálisan globálisan azonosítható, ha legföljebb egy algebrai halmaz elemeit kivéve minden más θ-ra $H_{\theta'}(s) = H_\theta(s)$-ből $\theta' = \theta$ következik. A lineáris rendszer akkor és csak akkor strukturálisan lokálisan azonosítható, ha a θ-t a $H_\theta(s)$ együtthatóinak számlálójába és nevezőjébe leképező függvény $(2n-1)km \times r$ -es Jacobi mátrixának rangja legföljebb egy algebrai halmaz elemeit kivéve minden más θ-ra a maximális r érték.

FIGYELMEZTETÉS: A rendkívül népszerű *hasonlósági transzformációs módszer* csak olyan általános idő-invariáns folytonos idejű véges dimenziós lineáris rendszerek esetében érvényes, ahol az állapot, az input és az output valós vektorokkal adható meg, és az A, B, C és D is valós együtthatós mátrixok. Ez a módszer kevéssé hasznos az olyan rendszereknél, amelyek állapot, input és output vektora pozitív. A segítségével levezethető legjobb kritérium a következő:

Hasonlósági transzformációs módszer. *A rendszer álljon legföljebb három szakaszból, és legyen legföljebb egy algebrai halmaz elemeinek kivételével minden paraméter-értékre strukturálisan vezérelhető és strukturálisan megfigyelhető. Ezen föltételek mellett a θ helyen akkor és csak akkor lesz a rendszer globálisan strukturálisan azonosítható, ha tetszőleges nemszinguláris T mátrixra a*

$$\begin{aligned} T\,A(\theta') &= A(\theta)\,T \\ T\,B(\theta') &= B(\theta) \\ C(\theta') &= C(\theta)\,T \end{aligned}$$

egyenletrendszernek létezik egyértelmű (I,θ) megoldása. Ha az egyenletrendszernek véges sok megoldása van, akkor a rendszer strukturálisan lokálisan azonosítható a θ helyen.

Négynél több szakaszú rendszerekre az előző kritérium elegendő, de nem szükséges. Ez a körülmény rendkívül behatárolja alkalmazhatóságát.

Az előző kritériumok esetében a számítógépes algebra nyújtja a legmegfelelőbb számítási eszközöket. A kadmium vándorlására vonatkozó példánkon fogjuk bemutatni ezeket a kritériumokat.

Először a **macro** lehetőségeit kihasználva az a_{ij}, b_{ij} és t_{ij} indexelt nevek helyett rövidítéseket vezetünk be. A **macro** argumentumainak kiértékelési szabályai miatt néhány trükköt is be kell vetnünk.

```
> sequence :=
>   seq(seq( a.i.j = a[i,j], i=0..3 ), j=0..3 ),
>   seq(seq( b.i.j = b[i,j], i=0..3 ), j=0..3 ),
>   seq(seq( t.i.j = t[i,j], i=0..3 ), j=0..3 ), c3 = c[3]:
> (eval@subs)( S=sequence, 'macro(S)' ):
```

Ezután a rendszert definiáló mátrixokat vezetjük be:

```
> A := matrix( 3, 3, [ -a21, a12, 0, a21,
> -(a02+a12+a32), a23, 0, a32, -(a03+a23) ] );
```

$$A := \begin{bmatrix} -a_{2,1} & a_{1,2} & 0 \\ a_{2,1} & -a_{0,2} - a_{1,2} - a_{3,2} & a_{2,3} \\ 0 & a_{3,2} & -a_{0,3} - a_{2,3} \end{bmatrix}$$

```
> B := matrix( 3, 1, [0,1,0] );
```

$$B := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

```
> C := matrix( 2, 3, [0,0,c3,0,a02,a03] );
```

$$C := \begin{bmatrix} 0 & 0 & c_3 \\ 0 & a_{0,2} & a_{0,3} \end{bmatrix}$$

```
> AB := evalm( A &* B ): A2B := evalm( A &* AB ):
> MC := augment( B, AB, A2B );
```

$$MC := \begin{bmatrix} 0 & a_{1,2} & -a_{2,1}\,a_{1,2} + a_{1,2}\,\%1 \\ 1 & \%1 & a_{2,1}\,a_{1,2} + \%1^2 + a_{2,3}\,a_{3,2} \\ 0 & a_{3,2} & a_{3,2}\,\%1 + (-a_{0,3} - a_{2,3})\,a_{3,2} \end{bmatrix}$$

$$\%1 := -a_{0,2} - a_{1,2} - a_{3,2}$$

Előttünk a csábító lehetőség: határozzuk meg az M_C vezérelhetőségi mátrix rangját a **rank** Maple eljárással.

```
> rank( MC );
```

$$3$$

Ez azonban csak a mátrix általános rangját adta meg arra az esetre, ha a mátrix együtthatóit a $\mathbb{Q}(a_{12}, a_{21}, \ldots, a_{32})$ hányadostest elemeinek tekintjük. Nekünk nem ez kell! A rang meghatározásának másik módja az **ffgausselim** eljárással végrehajtott törtmentes Gauss-elimináció.

```
> ffgausselim( MC );
```

$$\Big[1, -a_{0,2} - a_{1,2} - a_{3,2}, a_{2,1}\,a_{1,2} + a_{0,2}^2 + 2\,a_{1,2}\,a_{0,2} + 2\,a_{0,2}\,a_{3,2} + a_{1,2}^2 + 2\,a_{1,2}\,a_{3,2} + a_{3,2}^2 + a_{2,3}\,a_{3,2}\Big]$$
$$[\,0, a_{1,2}, -a_{2,1}\,a_{1,2} - a_{1,2}\,a_{0,2} - a_{1,2}^2 - a_{1,2}\,a_{3,2}\,]$$
$$[\,0, 0, -a_{1,2}\,a_{3,2}\,a_{0,3} - a_{1,2}\,a_{2,3}\,a_{3,2} + a_{3,2}\,a_{2,1}\,a_{1,2}\,]$$

```
> map( factor, " );
```

$$\Big[1, -a_{0,2} - a_{1,2} - a_{3,2}, a_{2,1}\,a_{1,2} + a_{0,2}^2 + 2\,a_{1,2}\,a_{0,2} + 2\,a_{0,2}\,a_{3,2} + a_{1,2}^2 + 2\,a_{1,2}\,a_{3,2} + a_{3,2}^2 + a_{2,3}\,a_{3,2}\Big]$$
$$[\,0, a_{1,2}, -a_{1,2}\,(a_{1,2} + a_{2,1} + a_{0,2} + a_{3,2})\,]$$
$$[\,0, 0, a_{1,2}\,a_{3,2}\,(-a_{0,3} - a_{2,3} + a_{2,1})\,]$$

Innen rögtön adódik a vezérelhetőség szükséges és elegendő föltétele: $a_{12} \neq 0$, $a_{32} \neq 0$ és $a_{21} \neq a_{03} + a_{23}$. Ne feledjük, hogy a Maple a mátrix együtthatóit a $\mathbb{Q}(a_{12}, a_{21}, \ldots, a_{32})$ hányadostest elemeinek tekinti, s ezért a törtmentes Gauss-elimináció bármely lépésénél előfordulhatna, hogy minden sort leoszt az előző főelemmel, vagy a sorban lévő polinomok legnagyobb közös osztójával. Az utóbbi műveletet nem implementálták a **gausselim** eljárásban, az előbbi meg nem okozhat problémát, mivel ezek a főelemek eleve nullától különbözőek. Ha nem kívánunk ezekre az implementáció-függő részletekre támaszkodni, használhatjuk a következő ekvivalens vezérelhetőségi kritériumot is.

A vezérelhetőség kritériuma. *A lineáris rendszer akkor és csak akkor vezérelhető, ha az M_C vezérelhetőségi mátrixra teljesül a* $\det(M_C M_C^T) \neq 0$ *(vagy négyzetes M_C esetén a* $\det(M_C) \neq 0$*) föltétel.*

```
> factor( det( MC ) );
```

$$-a_{1,2}\,a_{3,2}\,(-a_{0,3} - a_{2,3} + a_{2,1})$$

Ebből a kritériumból azonban legtöbbször bonyolultabb föltételeket kapunk.

A megfigyelhetőség ugyanígy ellenőrizhető. Föltesszük, hogy $a_{02} \neq 0$, hogy a C mátrix rangja 2 legyen.

```
> CA := evalm( C &* A ):
> MO := stack( C, CA );
```

$$MO :=$$
$$\begin{bmatrix} 0, 0, c_3 \end{bmatrix}$$
$$\begin{bmatrix} 0, a_{0,2}, a_{0,3} \end{bmatrix}$$
$$\begin{bmatrix} 0, c_3 a_{3,2}, c_3(-a_{0,3} - a_{2,3}) \end{bmatrix}$$
$$\Big[a_{0,2} a_{2,1}, a_{0,2}(-a_{0,2} - a_{1,2} - a_{3,2}) + a_{0,3} a_{3,2},$$
$$a_{0,2} a_{2,3} + a_{0,3}(-a_{0,3} - a_{2,3}) \Big]$$

```
> ffgausselim( MO );
```

$$\Big[a_{0,2} a_{2,1}, -a_{0,2}{}^2 - a_{1,2} a_{0,2} - a_{0,2} a_{3,2} + a_{0,3} a_{3,2},$$
$$a_{0,2} a_{2,3} - a_{0,3}{}^2 - a_{0,3} a_{2,3} \Big]$$
$$\begin{bmatrix} 0, a_{0,2}{}^2 a_{2,1}, a_{0,3} a_{0,2} a_{2,1} \end{bmatrix}$$
$$\begin{bmatrix} 0, 0, c_3 a_{0,2}{}^2 a_{2,1} \end{bmatrix}$$
$$[0, 0, 0]$$

Tehát a rendszer megfigyelhető, ha $a_{02} \neq 0$ és $a_{21} \neq 0$.

Példánknál a megfigyelhetőség következő kritériuma bonyolultabb, de ekvivalens föltételeket eredményez.

A megfigyelhetőség kritériuma. *A lineáris rendszer akkor és csak akkor megfigyelhető, ha az M_O megfigyelhetőségi mátrixra teljesül a $\det(M_O M_O^T) \neq 0$ (vagy négyzetes M_O esetén a $\det(M_O) \neq 0$) föltétel.*

```
> factor( det( transpose(MO) &* MO ) );
```

$$c_3{}^2 a_{2,1}{}^2 a_{0,2}{}^2 (2 a_{0,2} a_{3,2} a_{0,3} a_{2,3} + c_3{}^2 a_{3,2}{}^2 + a_{0,3}{}^2 a_{3,2}{}^2 + a_{0,2}{}^2$$
$$+ a_{0,2}{}^2 a_{0,3}{}^2 + a_{0,2}{}^2 a_{2,3}{}^2 + 2 a_{0,2}{}^2 a_{0,3} a_{2,3} + 2 a_{0,2} a_{3,2} a_{0,3}{}^2)$$

Úgy tűnik, mintha a negyedik tényezőből még egy extra föltétel következne. De ha ezt az a_{32} másodfokú polinomjának tekintjük, és kiszámítjuk a diszkriminánsát, kiderül, hogy mindig negatív értékeket vesz föl:

```
> collect( op(4,"), a32 );
```

$$(c_3{}^2 + a_{0,3}{}^2) a_{3,2}{}^2 + (2 a_{0,2} a_{0,3} a_{2,3} + 2 a_{0,2} a_{0,3}{}^2) a_{3,2}$$
$$+ a_{0,2}{}^2 a_{0,3}{}^2 + a_{0,2}{}^2 a_{2,3}{}^2 + 2 a_{0,2}{}^2 a_{0,3} a_{2,3} + a_{0,2}{}^2$$

```
> discrim(",a32);
```

$$-4 (c_3{}^2 + c_3{}^2 a_{0,3}{}^2 + c_3{}^2 a_{2,3}{}^2 + 2 c_3{}^2 a_{0,3} a_{2,3} + a_{0,3}{}^2) a_{0,2}{}^2$$

A továbbiakban a strukturális lokális azonosíthatóságot a Markov-féle paramétermátrix segítségével vizsgáljuk. Először a mátrixot és ennek Jacobi mátrixát kell megadnunk a Maple-nek.

```
> n := rowdim( A );
```

$$n := 3$$

```
> CA0B := evalm( C&*B ):
> for i from 1 to 5 do CA.i.B := evalm( C &* A^i &* B ) od:
> J := stack( seq(CA.i.B, i=0..2*n-1) ):
> J12 := map( diff, J, a12): J21 := map( diff, J, a21):
> J23 := map( diff, J, a23): J32 := map( diff, J, a32):
> J02 := map( diff, J, a02): J03 := map( diff, J, a03):
> JM := augment( J21, J12, J02, J23, J32, J03 ):
> ffgausselim( JM ):
> map( factor, " );
```

$$\Big[\%1\,,\, a_{0,2}\,a_{2,1} + 2\,a_{0,2}{}^2 + 2\,\%1 + 2\,\%3 - \%2\,,\, a_{2,1}\,a_{1,2} + 3\,a_{0,2}{}^2$$
$$+\,4\,\%1 + 4\,\%3 + a_{1,2}{}^2 + 2\,a_{1,2}\,a_{3,2} + a_{3,2}{}^2 + a_{2,3}\,a_{3,2} - \%2,$$
$$a_{3,2}\,(a_{0,2} - a_{0,3})\,,\, 2\,a_{0,2}{}^2 + 2\,\%1 + 2\,\%3 + a_{0,2}\,a_{2,3} - a_{0,3}\,a_{0,2}$$
$$-\,a_{0,3}\,a_{1,2} - 2\,\%2 - a_{0,3}{}^2 - a_{0,3}\,a_{2,3}\,,$$
$$-a_{3,2}\,(a_{3,2} + a_{0,2} + a_{1,2} + 2\,a_{0,3} + a_{2,3})\Big]$$
$$\Big[0\,,\, -a_{0,2}{}^2\,a_{1,2}\,,\, -(2\,a_{0,2} + a_{1,2} + a_{3,2})\,a_{0,2}\,a_{1,2}\,,\, 0\,,$$
$$-(a_{0,2} - a_{0,3})\,a_{0,2}\,a_{1,2}\,,\, a_{0,2}\,a_{1,2}\,a_{3,2}\Big]$$
$$[\,0\,,\, 0\,,\, -a_{0,2}{}^2\,a_{1,2}\,,\, 0\,,\, 0\,,\, 0\,]$$
$$\Big[\,0\,,\, 0\,,\, 0\,,\, c_3\,a_{0,2}{}^2\,a_{1,2}\,a_{3,2}\,,$$
$$c_3\,a_{0,2}\,a_{1,2}\,(a_{0,2}{}^2 + \%1 + a_{0,2}\,a_{2,3} + \%3 + a_{0,3}\,a_{0,2} + \%2)\,,$$
$$c_3\,a_{0,2}\,a_{1,2}\,a_{3,2}\,(a_{3,2} + a_{0,2})\,\Big]$$
$$[\,0\,,\, 0\,,\, 0\,,\, 0\,,\, c_3{}^2\,a_{3,2}\,a_{1,2}\,a_{0,2}{}^2\,,\, 0\,]$$
$$\Big[\,0\,,0\,,0\,,0\,,0\,,$$
$$c_3{}^2\,a_{0,2}{}^2\,a_{1,2}\,(-a_{0,3} + a_{2,1} - a_{2,3})\,a_{3,2}{}^2\,(a_{2,1} - a_{0,3} + a_{1,2} - a_{2,3})\,\Big]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$[\,0\,,0\,,0\,,0\,,0\,,0\,]$$
$$\%1 := a_{0,2}\,a_{1,2}$$
$$\%2 := a_{0,3}\,a_{3,2}$$
$$\%3 := a_{0,2}\,a_{3,2}$$

A rendszer akkor és csak akkor strukturálisan lokálisan azonosítható, ha a_{12}, a_{02} és a_{32} nemnulla, $a_{21} \neq a_{03} + a_{23}$ végül $a_{12} + a_{21} \neq a_{03} + a_{23}$. A strukturális globális azonosíthatóságot is megvizsgáljuk az átvitel-függvényes módszerrel.

```
> H := evalm( C &* (s-A)^(-1) &* B ):
```

```
> evalm( denom(H[1,1]) * H ) / collect(denom(H[1,1]),s);
```

$$[c_3\,(a_{2,1}+s)\,a_{3,2}]$$

$$\left[\%4\,\left(\frac{a_{0,2}\,(a_{2,1}+s)\,(a_{0,3}+a_{2,3}+s)}{\%4}+\frac{a_{0,3}\,(a_{2,1}+s)\,a_{3,2}}{\%4}\right)\right]\Big/$$

$$(s^3+(a_{2,1}+a_{0,2}+a_{1,2}+a_{0,3}+a_{2,3}+a_{3,2})\,s^2+(a_{2,1}\,a_{2,3}+a_{0,3}\,a_{1,2}$$
$$+\,a_{1,2}\,a_{2,3}+a_{0,3}\,a_{2,1}+a_{0,3}\,a_{3,2}+a_{0,2}\,a_{2,1}+a_{3,2}\,a_{2,1}+a_{0,3}\,a_{0,2}$$
$$+\,a_{0,2}\,a_{2,3})s+\%3+\%2+\%1)$$

$$\%1 := a_{2,1}\,a_{0,3}\,a_{0,2}$$

$$\%2 := a_{2,1}\,a_{0,2}\,a_{2,3}$$

$$\%3 := a_{0,3}\,a_{3,2}\,a_{2,1}$$

$$\%4 := \%3+s\,a_{0,3}\,a_{0,2}+s\,a_{0,3}\,a_{3,2}+a_{2,1}\,s\,a_{2,3}+s\,a_{0,3}\,a_{1,2}$$
$$+\,s\,a_{1,2}\,a_{2,3}+a_{2,1}\,s\,a_{0,3}+a_{2,1}\,s^2+a_{0,2}\,s^2+a_{1,2}\,s^2+a_{3,2}\,s^2$$
$$+\,s^2\,a_{0,3}+s^2\,a_{2,3}+s^3+\%2+\%1+a_{2,1}\,a_{0,2}\,s+a_{2,1}\,a_{3,2}\,s$$
$$+\,s\,a_{0,2}\,a_{2,3}$$

```
> collect( numer(H[1,1]), s );
```

$$c_3\,a_{3,2}\,s+c_3\,a_{2,1}\,a_{3,2}$$

```
> collect( numer(H[2,1]), s );
```

$$a_{0,2}\,s^2+(a_{0,2}\,a_{2,1}+a_{0,3}\,a_{0,2}+a_{0,2}\,a_{2,3}+a_{0,3}\,a_{3,2})\,s$$
$$+\,a_{2,1}\,(a_{0,3}\,a_{0,2}+a_{0,2}\,a_{2,3}+a_{0,3}\,a_{3,2})$$

Az átvitel-függvényes kritériumnak megfelelően a strukturális globális azonosíthatóság ekvivalens a következő utasításokkal generált egyenletrendszer egyértelmű megoldhatóságával:

```
> cfs1 := { coeffs( expand(numer(H[1,1])), s ) };
```

$$\mathit{cfs1} := \{c_3\,a_{3,2},\ c_3\,a_{2,1}\,a_{3,2}\}$$

```
> cfs2 := { coeffs( expand(numer(H[2,1])), s ) };
```

$$\mathit{cfs2} := \{a_{0,2}\,a_{2,1}+a_{0,3}\,a_{0,2}+a_{0,2}\,a_{2,3}+a_{0,3}\,a_{3,2},$$
$$a_{0,3}\,a_{3,2}\,a_{2,1}+a_{2,1}\,a_{0,2}\,a_{2,3}+a_{2,1}\,a_{0,3}\,a_{0,2},\ a_{0,2}\}$$

```
> cfs3 := { coeffs( expand(denom(H[1,1])), s ) }
> minus {1};
```

$$\mathit{cfs3} := \{a_{2,1}+a_{0,2}+a_{1,2}+a_{0,3}+a_{2,3}+a_{3,2},\,a_{2,1}\,a_{2,3}+a_{0,3}\,a_{1,2}$$
$$+\,a_{1,2}\,a_{2,3}+a_{0,3}\,a_{2,1}+a_{0,3}\,a_{3,2}+a_{0,2}\,a_{2,1}+a_{3,2}\,a_{2,1}+a_{0,3}\,a_{0,2}$$
$$+\,a_{0,2}\,a_{2,3},\ a_{0,3}\,a_{3,2}\,a_{2,1}+a_{2,1}\,a_{0,2}\,a_{2,3}+a_{2,1}\,a_{0,3}\,a_{0,2}\}$$

```
> cfs := `union`( cfs.(1..3) ):
> eqns := map( x -> x = subs( a12=b12, a21=b21, a23=b23,
>   a32=b32, a23=b23, a03=b03, a02=b02, x ), cfs );
```

$$eqns := \{c_3\, a_{2,1}\, a_{3,2} = c_3\, b_{2,1}\, b_{3,2},$$
$$a_{0,3}\, a_{3,2}\, a_{2,1} + a_{2,1}\, a_{0,2}\, a_{2,3} + a_{2,1}\, a_{0,3}\, a_{0,2} =$$
$$b_{0,3}\, b_{3,2}\, b_{2,1} + b_{2,1}\, b_{0,2}\, b_{2,3} + b_{2,1}\, b_{0,3}\, b_{0,2},\, a_{2,1}\, a_{2,3} + a_{0,3}\, a_{1,2}$$
$$+ a_{1,2}\, a_{2,3} + a_{0,3}\, a_{2,1} + a_{0,3}\, a_{3,2} + a_{0,2}\, a_{2,1} + a_{3,2}\, a_{2,1} + a_{0,3}\, a_{0,2}$$
$$+ a_{0,2}\, a_{2,3} = b_{2,1}\, b_{2,3} + b_{0,3}\, b_{1,2} + b_{1,2}\, b_{2,3} + b_{0,3}\, b_{2,1} + b_{0,3}\, b_{3,2}$$
$$+ b_{0,2}\, b_{2,1} + b_{3,2}\, b_{2,1} + b_{0,3}\, b_{0,2} + b_{0,2}\, b_{2,3},$$
$$a_{0,2}\, a_{2,1} + a_{0,3}\, a_{0,2} + a_{0,2}\, a_{2,3} + a_{0,3}\, a_{3,2} =$$
$$b_{0,2}\, b_{2,1} + b_{0,3}\, b_{0,2} + b_{0,2}\, b_{2,3} + b_{0,3}\, b_{3,2},\, a_{0,2} = b_{0,2},$$
$$a_{2,1} + a_{0,2} + a_{1,2} + a_{0,3} + a_{2,3} + a_{3,2} =$$
$$b_{2,1} + b_{0,2} + b_{1,2} + b_{0,3} + b_{2,3} + b_{3,2},\, c_3\, a_{3,2} = c_3\, b_{3,2}\}$$

```
> vars := {b12,b21,b23,b32,b02,b03}:
> solve(eqns,vars);
```

$$\{b_{2,1} = a_{2,1},\, b_{1,2} = a_{1,2},\, b_{2,3} = a_{2,3},\, b_{0,3} = a_{0,3},\, b_{0,2} = a_{0,2},$$
$$b_{3,2} = a_{3,2}\},\, \Big\{ b_{2,1} = a_{2,1}, b_{2,3} = (-a_{0,3}\, a_{0,2} - a_{0,3}\, a_{3,2} + a_{0,2}\, a_{2,1}$$
$$- a_{0,2}\, a_{2,3} + a_{0,2}\, a_{1,2} + a_{3,2}\, a_{2,1} + a_{1,2}\, a_{3,2}) \big/ a_{3,2},$$
$$b_{1,2} = a_{0,3} + a_{2,3} - a_{2,1},$$
$$b_{0,3} = -\frac{-a_{0,3}\, a_{0,2} - a_{0,3}\, a_{3,2} + a_{0,2}\, a_{2,1} - a_{0,2}\, a_{2,3} + a_{0,2}\, a_{1,2}}{a_{3,2}},$$
$$b_{0,2} = a_{0,2},\, b_{3,2} = a_{3,2} \Big\}$$

Két megoldás van. A rendszer tehát nem strukturálisan globálisan azonosítható, csupán strukturálisan lokálisan azonosítható.

Az utolsó eredményt a hasonlósági transzformációs módszerrel vezetjük le. Először az ugyanezen M modellnek megfelelő AA, BB és CC mátrixokat vezetjük be, csak most a_{12}, $a_{21}, \ldots$ helyett a b_{12}, $b_{21}, \ldots$ paraméterértékekkel számolunk.

```
> AA := subs( a12=b12, a21=b21, a23=b23, a32=b32,
> a23=b23, a03=b03, a02=b02, eval(A) );
```

$$AA := \begin{bmatrix} -b_{2,1} & b_{1,2} & 0 \\ b_{2,1} & -b_{0,2} - b_{1,2} - b_{3,2} & b_{2,3} \\ 0 & b_{3,2} & -b_{0,3} - b_{2,3} \end{bmatrix}$$

```
> BB := copy(B):
> CC := subs( a02=b02, a03=b03, eval(C) );
```

$$CC := \begin{bmatrix} 0 & 0 & c_3 \\ 0 & b_{0,2} & b_{0,3} \end{bmatrix}$$

```
> T := matrix( 3, 3, (i,j)->t[i,j] );
```

$$T := \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix}$$

Az alábbi kifejezés kiértékelésével

```
> evalm( T&*BB - B ):
```

$$\begin{bmatrix} t_{1,2} \\ t_{2,2} - 1 \\ t_{3,2} \end{bmatrix}$$

azt kapjuk, hogy $t_{12} = t_{32} = 0$ és $t_{22} = 1$. Ekkor a $CC - CT = 0$ mátrixegyenlet a következő alakra hozható:

```
> t12 := 0:  t32:= 0:  t22 := 1:
> evalm( CC - C&*T );
```

$$[\, -c_3\, t_{3,1} \,,\, 0 \,,\, c_3 - c_3\, t_{3,3} \,]$$
$$[\, -a_{0,2}\, t_{2,1} - a_{0,3}\, t_{3,1} \,,\, b_{0,2} - a_{0,2} \,,\, b_{0,3} - a_{0,2}\, t_{2,3} - a_{0,3}\, t_{3,3} \,]$$

```
> map( factor, " );
```

$$[\, -c_3\, t_{3,1} \,,\, 0 \,,\, -c_3\, (-1 + t_{3,3}) \,]$$
$$[\, -a_{0,2}\, t_{2,1} - a_{0,3}\, t_{3,1} \,,\, b_{0,2} - a_{0,2} \,,\, b_{0,3} - a_{0,2}\, t_{2,3} - a_{0,3}\, t_{3,3} \,]$$

Mivel $c_3 \neq 0$, rögtön következik, hogy $t_{31} = 0$, $t_{33} = 1$ és $b_{02} = a_{02}$.

```
> t31 := 0:  t33 := 1:  b02 := a02:
> evalm( CC - C&*T ): map( factor, " );
```

$$\begin{bmatrix} 0 & 0 & 0 \\ -a_{0,2}\, t_{2,1} & b_{0,2} - a_{0,2} & b_{0,3} - a_{0,2}\, t_{2,3} - a_{0,3} \end{bmatrix}$$

Ha $a_{02} \neq 0$ (ez teljesül, ha a rendszer megfigyelhető), akkor $t_{21} = 0$ és csak egy egyenletünk marad.

```
> t21 := 0:
> eqn1 := evalm(CC - C&*T)[2,3];
```

$$eqn1 := b_{0,3} - a_{0,2}\, t_{2,3} - a_{0,3}$$

Most levezetjük az A mátrix hasonlósági transzformációjából nyert nemtriviális egyenleteket:

```
> evalm( T&*AA - A&*T );
```

$$\left[-t_{1,1}\,b_{2,1}+a_{2,1}\,t_{1,1}\,,\ t_{1,1}\,b_{1,2}+t_{1,3}\,b_{3,2}-a_{1,2}\,,\ t_{1,3}\,(-b_{0,3}-b_{2,3})+a_{2,1}\,t_{1,3}-a_{1,2}\,t_{2,3}\right]$$

$$\left[b_{2,1}-a_{2,1}\,t_{1,1}\,,\ -b_{1,2}-b_{3,2}+t_{2,3}\,b_{3,2}+a_{1,2}+a_{3,2}\,,\ b_{2,3}+t_{2,3}\,(-b_{0,3}-b_{2,3})-a_{2,1}\,t_{1,3}-(-a_{0,2}-a_{1,2}-a_{3,2})\,t_{2,3}-a_{2,3}\right]$$

$$\left[0\,,\ b_{3,2}-a_{3,2}\,,\ -b_{0,3}-b_{2,3}-a_{3,2}\,t_{2,3}+a_{0,3}+a_{2,3}\right]$$

```
> eqn2 := map( op, convert(",'listlist') ):
> eqns := { eqn1, op(eqn2) } minus {0};
```

$$eqns := \{-t_{1,1}\,b_{2,1}+a_{2,1}\,t_{1,1},\ -b_{0,3}-b_{2,3}-a_{3,2}\,t_{2,3}+a_{0,3}+a_{2,3},\ -b_{1,2}-b_{3,2}+t_{2,3}\,b_{3,2}+a_{1,2}+a_{3,2},\ b_{2,3}+t_{2,3}\,(-b_{0,3}-b_{2,3})-a_{2,1}\,t_{1,3}-(-a_{0,2}-a_{1,2}-a_{3,2})\,t_{2,3}-a_{2,3},\ b_{0,3}-a_{0,2}\,t_{2,3}-a_{0,3},\ t_{1,1}\,b_{1,2}+t_{1,3}\,b_{3,2}-a_{1,2},\ t_{1,3}\,(-b_{0,3}-b_{2,3})+a_{2,1}\,t_{1,3}-a_{1,2}\,t_{2,3},\ b_{2,1}-a_{2,1}\,t_{1,1},\ b_{3,2}-a_{3,2}\}$$

```
> solve( eqns, {b12,b32,b23,b03,b21,t11,t13,t23} );
```

$$\{b_{2,1}=a_{2,1},\ b_{1,2}=a_{1,2},\ b_{2,3}=a_{2,3},\ b_{0,3}=a_{0,3},\ b_{3,2}=a_{3,2},\ t_{1,1}=1,\ t_{1,3}=0,\ t_{2,3}=0\},$$

$$\Big\{b_{2,1}=a_{2,1},\ b_{2,3}=(-a_{0,3}\,a_{0,2}-a_{0,3}\,a_{3,2}+a_{0,2}\,a_{2,1}-a_{0,2}\,a_{2,3}+a_{0,2}\,a_{1,2}+a_{3,2}\,a_{2,1}+a_{1,2}\,a_{3,2})\Big/a_{3,2},$$

$$b_{1,2}=a_{0,3}+a_{2,3}-a_{2,1},$$

$$b_{0,3}=-\frac{-a_{0,3}\,a_{0,2}-a_{0,3}\,a_{3,2}+a_{0,2}\,a_{2,1}-a_{0,2}\,a_{2,3}+a_{0,2}\,a_{1,2}}{a_{3,2}},$$

$$b_{3,2}=a_{3,2},\ t_{1,1}=1,\ t_{1,3}=\frac{-a_{0,3}-a_{2,3}+a_{2,1}+a_{1,2}}{a_{3,2}},$$

$$t_{2,3}=-\frac{-a_{0,3}-a_{2,3}+a_{2,1}+a_{1,2}}{a_{3,2}}\Big\}$$

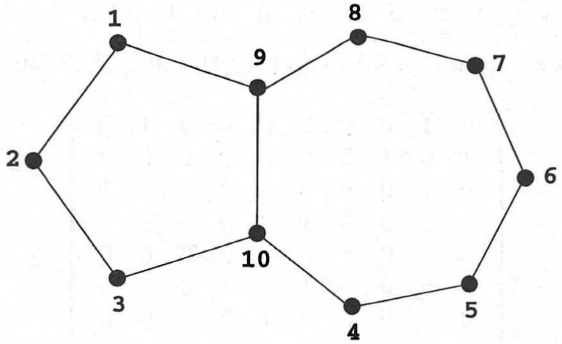
Két különböző megoldást találtunk, ami ismét azt bizonyítja, hogy a rendszer nem strukturálisan globálisan azonosítható, csak strukturálisan lokálisan azonosítható.

Bonyolultabb rendszerek esetében a vezérelhetőség, a megfigyelhetőség és a strukturális azonosíthatóság az itt vázolt eszközökkel már nem ilyen könnyedén tanulmányozható. Megfelelő algebrai egyenletmegoldó módszereket (a karakterisztikus halmazok vagy a Gröbner bázisok módszerét) kell bevetnünk a válaszok megkereséséhez. De legalább a számolási munkákat a számítógépes algebrai rendszer végzi a lineáris algebrai, az egyenletmegoldó stb. . csomagjaival.

19.3. A molekulapályák Hückel-elmélete

A következő példa, a π-elekton energiáknak és a molekulák elektromos töltéseloszlásának kiszámítására alkalmazható Hückel-elmélet a kvantumkémiából származik. Mivel a módszer megtalálható bármely kvantumkémiai szakkönyvben, például a [24]-ben is, ezért csak röviden vázoljuk.

Konkrétan a $C_{10}H_{10}$ képletű azulénmolekulát fogjuk vizsgálni, amelynek tíz szénatomból álló vázát a 19.2. ábrának megfelelő úgynevezettt σ-kötések kapcsolják össze.



19.2. ábra: Az azulén váza

Minden σ-kötés két elektront tartalmaz, ezért még 10 darab π-elektronnak nevezett elektronunk marad. Ezek a π-elektronok π-kötéseket alkotnak, amelyek a szénatomok $2p_z$ elektronpályáiból jönnek létre. A molekula-pályákat ezekkel az atompályákkal $\psi = c_1\phi_1 + c_2\phi_2 + \cdots + c_{10}\phi_{10}$ alakú lineáris kombinációkként írjuk föl. A $c_1, c_2, \ldots, c_{10}$ együtthatókat variáció-számítással abból a föltételből határozzuk meg, hogy az energia a lehető legkisebb legyen. Ez a következő általánosított sajátérték-problémához vezet: $(H - ES)C = 0$, ahol H a Hamilton-mátrix a $< \phi_i|H|\phi_j >$ elemekkel, az S átfedési mátrix elemeit $< \phi_i|\phi_j >$ adja, C pedig általánosított sajátvektor az E általánosított sajátértékkel. A Hückel-elméletben a következő föltételezésekkel szokás élni:

- Minden H_{ii} mátrixelem azonos, α-val jelölt érték.
- Ha az i-dik és a j-dik atomok szomszédosak, az összes megfelelő H_{ij} megegyezik, ezt a közös értéket jelölje β. Ha az i-dik és a j-dik atom nem kapcsolódik közvetlenül egymáshoz, akkor $H_{ij} = 0$.
- Az átfedés elhanyagolható, vagyis S a zérusmátrix.

Ezen föltételek mellett közönséges sajátérték-problémát kapunk. Az azulénhez

tartozó szimmetrikus H mátrix (a molekula Hückel mátrixa) a következő:

$$\begin{pmatrix} \alpha & \beta & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 \\ \beta & \alpha & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & \beta \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 & 0 & \beta \\ 0 & 0 & 0 & \beta & \alpha & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta & \alpha & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta & \alpha & \beta & 0 \\ \beta & 0 & 0 & 0 & 0 & 0 & 0 & \beta & \alpha & \beta \\ 0 & 0 & \beta & \beta & 0 & 0 & 0 & 0 & \beta & \alpha \end{pmatrix}$$

A Hückel mátrix sajátérték-problémája helyett a megfelelő topologikus mátrix, vagyis

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

sajátérték-problémáját oldjuk meg. A topologikus mátrix valamely x sajátértéke és az E sajátérték között egyszerű kapcsolat van: $E = \alpha + \beta x$, a megfelelő sajátvektorok pedig megegyeznek.

Kezdjük el a számolást. Először fölsoroljuk a közvetlen kapcsolatban lévő szénatom-párokat:

```
> 'number of C-atoms' := 10:
> Azulene_skeleton :=
>   [ [1,2], [1,9], [2,3], [3,10], [4,5], [4,10],
>   [4,10], [5,6], [6,7], [7,8], [8,9],[9,10] ]:
```

Ezután ritka szimmetrikus mátrixként definiáljuk a Hückel mátrixot:

```
> Hueckel_matrix := array( sparse,
>   1..'number of C-atoms', 1..'number of C-atoms' ):
> for i from 1 to nops( Azulene_skeleton ) do
>   Hueckel_matrix[ op( Azulene_skeleton[i] ) ] := beta
> od:
> Hueckel_matrix := evalm( alpha + Hueckel_matrix
>   + transpose( Hueckel_matrix ) ):
> topological_matrix := subs( alpha=0, beta=1,
>   eval( Hueckel_matrix ) ):
```

A topologikus mátrix karakterisztikus polinomja a `linalg` csomag **charpoly** eljárásával, a karakterisztikus mátrix pedig a **charmat** rutinnal határozható meg.

```
> factor( charpoly( topological_matrix, x ) );
```

$$(x^4+x^3-3\,x^2-x+1)\,(x^6-x^5-7\,x^4+5\,x^3+13\,x^2-6\,x-4)$$

```
> charmat( topological_matrix, x );
```

$$\begin{bmatrix} x & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & x & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & x & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & x & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & x & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & x & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & x & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & x & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & x & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & x \end{bmatrix}$$

```
> factor( det(") );
```

$$(x^4+x^3-3\,x^2-x+1)\,(x^6-x^5-7\,x^4+5\,x^3+13\,x^2-6\,x-4)$$

A sajátértékek és a sajátvektorok az **eigenvalues**-zal, illetve az **eigenvectors**-zal számíthatók ki.

```
> eigenvalues( topological_matrix );
```

$$\mathrm{RootOf}(_Z^4+_Z^3-3\,_Z^2-_Z+1),\\ \mathrm{RootOf}(_Z^6-_Z^5-7\,_Z^4+5\,_Z^3+13\,_Z^2-6\,_Z-4)$$

```
> eigenvectors( topological_matrix );
```

$$[\%2,1,\{[\cdots]\}],\ [\%1,1,\{[\cdots]\}],\\ \%1 := \mathrm{RootOf}(_Z^4+_Z^3-3\,_Z^2-_Z+1)\\ \%2 := \mathrm{RootOf}(_Z^6-_Z^5-7\,_Z^4+5\,_Z^3+13\,_Z^2-6\,_Z-4)$$

A sajátvektorokat mutató eredmény egy részét elhagytuk. Tekintsük az előző sorozat utolsó elemét:

```
> "[2];
```

$$[\%1,1,\{[-1,0,1,\%1^2+\%1-1,\%1^3+\%1^2-2\,\%1,0,\\ -\%1^3-\%1^2+2\,\%1,-\%1^2-\%1+1,-\%1,\%1]\}]\\ \%1 := \mathrm{RootOf}(_Z^4+_Z^3-3\,_Z^2-_Z+1)$$

Az első komponens a sajátvektorhoz tartozó sajátértéket, a második a sajátérték multiplicitását, a harmadik pedig magát a sajátvektort adja meg. Ha a sajátaltér többdimenziós, a harmadik komponens ennek egy bázisát tartalmazza. Az előző objektumot tulajdonképpen négy sajátvektor leírásának kell tekintenünk, mivel texttt%1 négy értéket vehet föl; egy vektoron belül persze mindig ugyanazt az értékét kell tekinteni. Az alábbiakban a $-\frac{1}{4}+\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{11-\sqrt{5}\sqrt{2}}$ sajátértéknek megfelelő esetet láthatjuk.

```
> allvalues(",'dependent')[1];
```

$$[\%1, 1, \{\Big[-1, 0, 1, \%1^2-\frac{5}{4}+\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{22-2\sqrt{5}},$$
$$\%1^3+\%1^2+\frac{1}{2}-\frac{1}{2}\sqrt{5}-\frac{1}{2}\sqrt{22-2\sqrt{5}}, 0,$$
$$-\%1^3-\%1^2-\frac{1}{2}+\frac{1}{2}\sqrt{5}+\frac{1}{2}\sqrt{22-2\sqrt{5}},$$
$$-\%1^2+\frac{5}{4}-\frac{1}{4}\sqrt{5}-\frac{1}{4}\sqrt{22-2\sqrt{5}}, \frac{1}{4}-\frac{1}{4}\sqrt{5}-\frac{1}{4}\sqrt{22-2\sqrt{5}}, \%1\Big]\}]$$
$$\%1 := -\frac{1}{4}+\frac{1}{4}\sqrt{5}+\frac{1}{4}\sqrt{22-2\sqrt{5}}$$

Az M mátrix szinguláris értékei, vagyis az MM^T sajátértékeinek négyzetgyökei a **singularvals** eljárással számíthatók ki. Numerikus sajátértékeket és sajátvektorokat az **Eigenvals**-sal határozhatunk meg.

```
> eigenvalues := evalf(
>    Eigenvals( topological_matrix, 'eigenvectors' ) );
```

$$eigenvalues := [-2.095293985, -1.869213984, -1.579218099, -.7376403029, -.4003923188, .4772599976, .8869752420, 1.355674293, 1.651572314, 2.310276842]$$

Ellenőrizzük az első sajátvektort:

```
> subvector( eigenvectors, 1..`number of C-atoms`, 1 );
```

$$[-.2590786300, .1881\,10^{-8}, .2590786280, .3354972117, -.1601193962, -.2190\,10^{-8}, .1601194004, -.3354972138, .5428458950, -.5428458923]$$

```
> evalm( ( topological_matrix
>    - eigenvalues[1] ) &* " );
```

$$[.18\,10^{-8}, .19\,10^{-8}, .5\,10^{-9}, .12\,10^{-8}, .1810\,10^{-8}, -.4\,10^{-9}, .5\,10^{-9}, .13\,10^{-8}, .29\,10^{-8}, .2\,10^{-8}]$$

```
> map( fnormal, " ); # floating-point normalization
```

$$[\,0, 0, 0, 0, 0, 0, 0, 0, 0, 0\,]$$

Példánkat a π-elektron energiájának és sűrűségfüggvényének numerikus leírásával zárjuk. Ha az ortonormált ψ_i molekula-pályákon egyenként n_i darab π-elektron van, akkor a j-dik szénatom q_j π-elektron-sűrűsége:

$$q_j = \sum_{i=0}^{10} n_i c_{ji}^2.$$

A továbbiakban a legkisebb π-elektron energiának megfelelő állapotra koncentrálunk.

```
> occupation_numbers := [0,0,0,0,0,2,2,2,2,2,2]:
> pi_electron_energy := sum( 'occupation_numbers[i]
>   * ( alpha + eigenvalues[i]*beta )',
>   'i'=1..`number of C-atoms` );
```

$$pi_electron_energy := 10\,\alpha + 13.36351737\,\beta$$

Nincsenek többszörös sajátértékek. (Egyébként még Gram-Schmidt ortogonalizációt kellett volna végeztetnünk a `linalg` csomag **GramSchmidt** eljárásával.) Már csak normalizálnunk kell a sajátvektorokat. A **normalize** éppen megfelel erre a célra.

```
> eigenvectors := map( normalize, [ seq(
>   subvector( eigenvectors, 1..`number of C-atoms`, i ),
>   i=1..`number of C-atoms` ) ] ):
```

Minden készen áll a π-elektronok sűrűségének kiszámításához.

```
> electron_density := seq( sum( 'occupation_numbers[i] *
>   (eigenvectors[i][j])^2', 'i'=1..`number of C-atoms` ),
>   j=1..`number of C-atoms` );
```

$$electron_density := 1.172879340, 1.046599708, 1.172879339,\\ .8549456656, .9864468455, .8700013802, .9864468483,\\ .8549456632, 1.027427606, 1.027427605$$

A vegyészeknek ez az egyszerű modell is megmagyarázza a helyettesítések helyfüggőségét. Az elektrofil helyettesítések (például egy klóratomé) legnagyobb valószínűséggel a legmagasabb elektronsűrűségeknek megfelelő pozíciókon mennek végbe. Az azulén esetében ez a 2. pozíció. A nukleofil helyettesítések (például egy metilgyök) legvalószínűbb helyét a legalacsonyabb elektronsűrűségek adják, ez most a 4. és a 8. pozíciót jelenti.

19.4. Vektoranalízis

A `linalg` csomagban megtalálhatók az olyan vektoranalízishez kapcsolódó eljárások, mint a **grad** (a gradiensre), a **diverge** (a divergenciára), a **laplacian**, a **hessian** és a **curl** (a rotációra). Mindezek jól alkalmazhatók különböző ortogonális görbevonalú koordinátarendszerek esetén is.

```
> alias( f=f(x,y,z), g=g(x,y,z), h=h(x,y,z), c=[x,y,z] ):
> grad( f, c );
```

$$\left[\frac{\partial}{\partial x} f, \frac{\partial}{\partial y} f, \frac{\partial}{\partial z} f\right]$$

```
> diverge( [f,g,h], c );
```

$$(\frac{\partial}{\partial x} f) + (\frac{\partial}{\partial y} g) + (\frac{\partial}{\partial z} h)$$

```
> laplacian( f, c );
```

$$(\frac{\partial^2}{\partial x^2} f) + (\frac{\partial^2}{\partial y^2} f) + (\frac{\partial^2}{\partial z^2} f)$$

```
> jacobian( [f,g,h], c );
```

$$\begin{bmatrix} \frac{\partial}{\partial x} f & \frac{\partial}{\partial y} f & \frac{\partial}{\partial z} f \\ \frac{\partial}{\partial x} g & \frac{\partial}{\partial y} g & \frac{\partial}{\partial z} g \\ \frac{\partial}{\partial x} h & \frac{\partial}{\partial y} h & \frac{\partial}{\partial z} h \end{bmatrix}$$

```
> hessian( f, c );
```

$$\begin{bmatrix} \frac{\partial^2}{\partial x^2} f & \frac{\partial^2}{\partial y\, \partial x} f & \frac{\partial^2}{\partial z\, \partial x} f \\ \frac{\partial^2}{\partial y\, \partial x} f & \frac{\partial^2}{\partial y^2} f & \frac{\partial^2}{\partial z\, \partial y} f \\ \frac{\partial^2}{\partial z\, \partial x} f & \frac{\partial^2}{\partial z\, \partial y} f & \frac{\partial^2}{\partial z^2} f \end{bmatrix}$$

```
> curl( [f,g,h], c );
```

$$\left[(\frac{\partial}{\partial y} h) - (\frac{\partial}{\partial z} g), (\frac{\partial}{\partial z} f) - (\frac{\partial}{\partial x} h), (\frac{\partial}{\partial x} g) - (\frac{\partial}{\partial y} f)\right]$$

```
> diverge( ", c );
```

$$0$$

```
> curl( grad( f, c ), c );
```

$$[\,0\;0\;0\,]$$

A két utolsó parancs eredménye jól ismert tulajdonságokat fejez ki. A 19.1. és a 19.2. táblázatban soroltuk föl azokat a legfontosabb ortogonális görbevonalú koordinátarendszereket, amelyek használatakor a Maple beépített lehetőségeinek segítségével ki tudja fejezni a gradienst, a divergenciát, a Laplace-operátort és a rotációt. A beépített koordinátarendszerekről teljes listát és részletesebb magyarázatot a **?changecoords** paranccsal kérhetünk. A koordinátarendszerek definíciója megtalálható a [142, 171] irodalmakban.

Koordináták	Definíció	Maple elnevezés
derékszögű	(u, v)	cartesian
polár	$(u\cos v, u\sin v)$	polar
bipolár	$\left(\dfrac{\sinh v}{\cosh v - \cos u}, \dfrac{\sin u}{\cosh v - \cos u}\right)$	bipolar
elliptikus	$(\cosh u\cos v, \sinh u\sin v)$	elliptic
parabolikus	$\left(\dfrac{u^2 - v^2}{2}, uv\right)$	parabolic
hiperbolikus	$\left(\sqrt{u + \sqrt{u^2 + v^2}}, \sqrt{-u + \sqrt{u^2 + v^2}}\right)$	hyperbolic

19.1. táblázat: Előre definiált 2D koordinátarendszerek

Koordináták	Definíció	Maple elnevezés
derékszögű	(u, v, w)	cartesian
kör-henger	$(u\cos v, u\sin v, w)$	cilindrical
elliptikus-henger	$(a\cosh u\cos v,$ $a\cosh u\sin v, w)$	ellcilindrical
parabolikus-henger	$(uv\cos w, uv\sin w, \frac{u^2-v^2}{2})$	paraboloidal
hiperbolikus-henger	$(\sqrt{u + \sqrt{u^2 + v^2}},$ $\sqrt{v + \sqrt{u^2 + v^2}},$ $w)$	hypercilindrical
szférikus	$(u\cos v\sin w,$ $u\sin v\sin w,$ $u\cos v)$	spherical
nyújtott szferodikus	$(a\sinh u\sin v\cos w,$ $a\sinh u\sin v\sin w,$ $a\cosh u\cos v)$	prolatespheroidal
összenyomott szferodikus	$(a\cosh u\sin v\cos w,$ $a\cosh u\sin v\sin w,$ $a\sinh u\cos v)$	oblatespheroidal
ellipszodikus	$(\frac{uvw}{ab},$ $\frac{1}{b}\sqrt{\frac{(u^2-b^2)(v^2-b^2)(b^2-w^2)}{a^2-b^2}},$ $\frac{1}{a}\sqrt{\frac{(u^2-a^2)(a^2-v^2)(a^2-w^2)}{a^2-b^2}},$	ellipsoidal

19.2. táblázat: Előre definiált 3D koordinátarendszerek

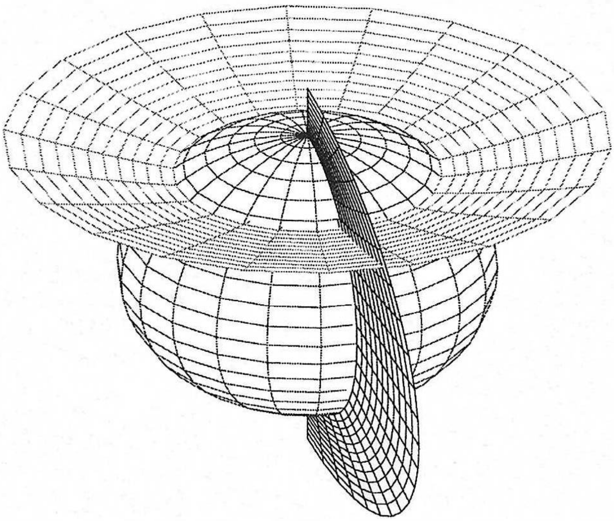
Példaként vegyük a Maple-ben a következő formulákkal definiált nyújtott szferodikus koordinátákat:

$$\begin{aligned} x &= a \sinh u \sin v \cos w \\ y &= a \sinh u \sin v \sin w \\ z &= a \cosh u \cos v \end{aligned}$$

Az ezeknek megfelelő koordináta-transzformációt a **plots[coordplot3d]** függvénnyel tudjuk megjeleníteni.

```
> plots[coordplot3d]( prolatespheroidal );
```

A grafikon a 19.3. ábrán látható.



19.3. ábra: A nyújtott szferodikus koordinátákra való transzformálás grafikonja

A gradiens, a rotáció, a divergencia és a Laplace-operátor könnyen kiszámolható, ha a megfelelő Maple parancsokat a **coords** = *coordinate system* opcióval egészítjük ki.

```
> alias( f=f(u,v,w), g=g(u,v,w), h=h(u.v.w), c=[u,v,w] ):
> grad( f, c, coords = prolatespheroidal(a) );
```

$$\left[\frac{\frac{\partial}{\partial u} f}{a\sqrt{\sinh(u)^2+\sin(v)^2}}, \frac{\frac{\partial}{\partial v} f}{a\sqrt{\sinh(u)^2+\sin(v)^2}}, \frac{\frac{\partial}{\partial w} f}{a\sinh(u)\sin(v)}\right]$$

Ha a számolás elkezdése előtt szeretnénk beállítani a koordinátarendszert, hogy az egyes parancsokban nem kelljen mindig megismételni a koordinátarendszer nevét, érdemes egy pillantást vetni a **plots[changecoords]** rutin forráskódjára. Ezt utánozva megírhatjuk az alábbihoz hasonló **setcoords** eljárást:

```
> interface( verboseproc = 2 ): # make Maple communicative
> readlib( 'plots/changecoords' ): # load library function
> print( 'plots/changecoords' );

proc(p, coord)
local c_name, a, b, c;
option
'Copyright (c) 1994 by the University of Waterloo'
    a := 1;
    b := 1/2;
    c := 1/3;
    if type(coord, function) then
     c_name := op(0, coord);
     if nops(coord) = 1 then a := op(1, coord)
     elif nops(coord) = 2 then a:= op(1,coord); b:= op(2,coord)
     elif nops(coord) = 3 then
       a := op(1, coord); b := op(2, coord); c := op(3, coord)
     else ERROR('Inappropriate number of parameters.')
    fi
    else c_name := coord
    fi;
    if member(c_name, readlib('plot3d/coordset2')()) then
        'plots/changecoords/twotrans'(p, c_name, a)
    elif member(c_name, readlib('plot3d/coordset')()) then
        'plots/changecoords/threetrans'(p, c_name, a, b, c)
    else ERROR('Cannot convert to coordinate system', coord)
    fi
end
```

Ez a Maple kód könnyen adaptálható:

```
> setcoords := proc( coord )
>   local c_name;
>   global curl, diverge, grad, laplacian,
>       scalefactors, 'setcoords/cname';
>   if type(coord, function) then
>       c_name := op(0,coord);
>     if nops(coord) > 3 then
>       ERROR('Inappropriate number of parameters.')
>     fi
>   else c_name := coord
>   fi;
>   if member(c_name,readlib('plot3d/coordset2')()) or
>     member(c_name,readlib('plot3d/coordset')())
>   then
>     'setcoords/cname' := coord,
>     unprotect( curl, diverge, grad, laplacian,
>       scalefactors ):
>     curl := proc() linalg[curl]( args,
>       coords = 'setcoords/cname' ) end;
```

```
>       diverge := proc() linalg[diverge]( args,
>         coords = 'setcoords/cname' ) end;
>       grad := proc() linalg[grad]( args,
>         coords = 'setcoords/cname' ) end;
>       laplacian := proc() linalg[laplacian]( args,
>         coords = 'setcoords/cname' ) end;
>       scalefactors := proc() readlib('linalg/scalefcts')(
>         args, 'setcoords/cname') end;
>       protect( curl, diverge, grad, laplacian,
>         scalefactors )
>     else
>       ERROR('Cannot convert to coordinate system',coord)
>     fi
>   end:
```

Minden használatra készen áll. Még egy olyan rutint is fölvettünk, amely a skálafaktorokat állítja be. (Ez eléggé furcsa módon el van rejtve a **linalg** csomagban.) A skálafaktorok fölhasználhatók például az új koordinátarendszernek megfelelő térfogatelem meghatározására.

```
> setcoords( prolatespheroidal(a) ):
> laplacian( f, c );
```

$$\Big(a\cosh(u)\sin(v)\,(\frac{\partial}{\partial u}f) + a\sinh(u)\sin(v)\,(\frac{\partial^2}{\partial u^2}f) + a\sinh(u)\cos(v)\,(\frac{\partial}{\partial v}f) + a\sinh(u)\sin(v)\,(\frac{\partial^2}{\partial v^2}f) + \frac{a\,(\sinh(u)^2+\sin(v)^2)\,(\frac{\partial^2}{\partial w^2}f)}{\sinh(u)\sin(v)}\Big)\Big/ \Big(a^3\,(\sinh(u)^2+\sin(v)^2)\sinh(u)\sin(v)\Big)$$

```
> scalefactors( c );
```

$$[a\sqrt{\sinh(u)^2+\sin(v)^2},\ a\sqrt{\sinh(u)^2+\sin(v)^2},\ a\sinh(u)\sin(v)]$$

```
> dV := convert( ", '*' ); # the volume element
```

$$dV := a^3\,(\sinh(u)^2+\sin(v)^2)\sinh(u)\sin(v)$$

Nyújtott szferodikus koordinátákat használnak a Slater–Zener típusú atompályák átfedési integráljainak kiszámításakor. (V. ö. [24].) Ennél az alkalmazásnál a koordináták szokásos jelölése ξ, η és ϕ, definíciójuk:

$$\begin{aligned} x &= a\sqrt{(\xi^2-1)(1-\eta^2)}\cos\phi, \\ y &= a\sqrt{(\xi^2-1)(1-\eta^2)}\sin\phi, \\ z &= a\xi\eta, \end{aligned}$$

ahol

$$0 < a,\ 1 \le \xi \le \infty,\ -1 \le \eta \le 1,\ 0 \le \phi \le 2\pi.$$

Ezt az új definíciót az **addcoords** eljárással tudjuk földolgoztatni.

```
> readlib( addcoords ):  # load the library function
> addcoords(
>   myprolatespheroidal, # name of coordinate system
>   [xi,eta,phi], # list of variables
>   [a*sqrt((xi^2-1)*(1-eta^2))*cos(phi),
>   a*sqrt((xi^2-1)*(1-eta^2))*sin(phi),
>   a*xi*eta ], # the cartesian coordinates x,y,z
>             # in terms of the new coordinates
>   [a], # list of names of constants
>   [[1.1],[sqrt(2)/2],[Pi/3], # default values for
>    [1..1.2,-1..1,0..2*Pi], # coordplot or
>    [-1.2..1.2,-1.2..1.2,-1.7..1.7]] # coordplot3d
> ):
```

Az eredmények kicsit bonyolultak, mivel a Maple **signum** és **abs** rutinjai nem mindig képesek a helyes előjelek fölismerésére. De könnyen kisegíthetjük a rendszert.

```
> assume( a>0, xi>=1, -1<=eta, eta<=1, 0<=phi, phi<=2*Pi );
> alias( f = f(xi,eta,phi), c = [xi,eta,phi] ):
> setcoords( myprolatespheroidal(a) ):
> grad( f, c );
```

$$\Big[\frac{\partial}{\partial \xi^{\sim}} f \Big/ (a^{\sim 2}$$
$$(\xi^{\sim 2} - 2\,\xi^{\sim 2}\,\eta^{\sim 2} + \xi^{\sim 2}\,\eta^{\sim 4} + \eta^{\sim 2}\,\sqrt{-(\xi^{\sim 2} - 1)\,(-1 + \eta^{\sim 2})}\,\sqrt{\%1})$$
$$\Big/ (\sqrt{-(\xi^{\sim 2} - 1)\,(-1 + \eta^{\sim 2})}\,\sqrt{\%1}))^{1/2},\ \frac{\partial}{\partial \eta^{\sim}} f \Big/ (a^{\sim 2}$$
$$(\eta^{\sim 2}\,\xi^{\sim 4} - 2\,\xi^{\sim 2}\,\eta^{\sim 2} + \eta^{\sim 2} + \xi^{\sim 2}\,\sqrt{-(\xi^{\sim 2} - 1)\,(-1 + \eta^{\sim 2})}\,\sqrt{\%1})$$
$$\Big/ (\sqrt{-(\xi^{\sim 2} - 1)\,(-1 + \eta^{\sim 2})}\,\sqrt{\%1}))^{1/2},$$
$$\frac{1}{2}\,\frac{\sqrt{4}\,(\frac{\partial}{\partial \phi^{\sim}} f)}{\sqrt{a^{\sim 2}\,\sqrt{-(\xi^{\sim 2} - 1)\,(-1 + \eta^{\sim 2})}\,\sqrt{\%1}}}\Big]$$
$$\%1 := |\xi^{\sim 2} - \xi^{\sim 2}\,\eta^{\sim 2} - 1 + \eta^{\sim 2}|$$

```
> map( simplify@combine@factor, " );
```

$$\Big[\frac{(\xi^{\sim} - 1)\,(\xi^{\sim} + 1)\,(\frac{\partial}{\partial \xi^{\sim}} f)}{a^{\sim}\,\sqrt{(-\eta^{\sim} + \xi^{\sim})\,(\eta^{\sim} + \xi^{\sim})\,(\xi^{\sim} - 1)\,(\xi^{\sim} + 1)}},$$
$$-\frac{(\eta^{\sim} - 1)\,(\eta^{\sim} + 1)\,(\frac{\partial}{\partial \eta^{\sim}} f)}{a^{\sim}\,\sqrt{(-\eta^{\sim} + \xi^{\sim})\,(\eta^{\sim} + \xi^{\sim})\,(-\eta^{\sim} + 1)\,(\eta^{\sim} + 1)}},$$
$$-\frac{\sqrt{\xi^{\sim 2} - 1}\,\sqrt{1 - \eta^{\sim 2}}\,(\frac{\partial}{\partial \phi^{\sim}} f)}{a^{\sim}\,(\xi^{\sim} - 1)\,(\xi^{\sim} + 1)\,(\eta^{\sim} - 1)\,(\eta^{\sim} + 1)}\Big]$$

```
> scalefactors( c ):
> map( simplify@combine@factor, " );
```

$$[a^{\sim}\sqrt{\frac{(-\eta^{\sim}+\xi^{\sim})(\eta^{\sim}+\xi^{\sim})}{(\xi^{\sim}-1)(\xi^{\sim}+1)}},\ a^{\sim}\sqrt{\frac{(-\eta^{\sim}+\xi^{\sim})(\eta^{\sim}+\xi^{\sim})}{(-\eta^{\sim}+1)(\eta^{\sim}+1)}},\ a^{\sim}\sqrt{\xi^{\sim 2}-1}\sqrt{1-\eta^{\sim 2}}]$$

```
> dV := simplify( convert( ", '*' ) );
```

$$dV := -a^{\sim 3}(\eta^{\sim}-\xi^{\sim})(\eta^{\sim}+\xi^{\sim})$$

Természetesen az eredmények jobban kézben tarthatók, ha kisebb lépésekben, a részeredmények egyszerűsítésével végezzük a számításokat. Kezdhetjük például a továbbiakban T-vel jelölt koordináta-transzformáció J Jacobi mátrixának kiszámításával is.

```
> restart; with(linalg):

Warning, new definition for norm

Warning, new definition for trace

> assume( a>0, xi>=1, -1<=eta, eta<=1, 0<=phi, phi<=2*Pi );
> alias( f = f(xi,eta,phi), c = [xi,eta,phi] ):
> T := [a*sqrt((xi^2-1)*(1-eta^2))*cos(phi),
> a*sqrt((xi^2-1)*(1-eta^2))*sin(phi), a*xi*eta ];
```

$$T := [a^{\sim}\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}\cos(\phi^{\sim}),\ a^{\sim}\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}\sin(\phi^{\sim}),\ a^{\sim}\xi^{\sim}\eta^{\sim}]$$

```
> J := jacobian(T,[xi,eta,phi]);
```

$$J := \left[\begin{array}{l}
\left[\frac{a^{\sim}\cos(\phi^{\sim})\xi^{\sim}(1-\eta^{\sim 2})}{\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}},-\frac{a^{\sim}\cos(\phi^{\sim})(\xi^{\sim 2}-1)\eta^{\sim}}{\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}},-a^{\sim}\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}\sin(\phi^{\sim})\right]\\
\left[\frac{a^{\sim}\sin(\phi^{\sim})\xi^{\sim}(1-\eta^{\sim 2})}{\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}},-\frac{a^{\sim}\sin(\phi^{\sim})(\xi^{\sim 2}-1)\eta^{\sim}}{\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}},a^{\sim}\sqrt{(\xi^{\sim 2}-1)(1-\eta^{\sim 2})}\cos(\phi^{\sim})\right]\\
[a^{\sim}\eta^{\sim},\ a^{\sim}\xi^{\sim},\ 0]
\end{array}\right]$$

A *g metrikus tenzor* $J^T J$-vel egyenlő.

```
> g := evalm( transpose(J) &* J ): g := map(simplify,g);
```

$$g := \begin{bmatrix} -\dfrac{a^{\sim 2}(-\xi^{\sim 2}+\eta^{\sim 2})}{\xi^{\sim 2}-1}, 0, 0 \\ 0, \dfrac{a^{\sim 2}(-\xi^{\sim 2}+\eta^{\sim 2})}{-1+\eta^{\sim 2}}, 0 \\ 0, 0, a^{\sim 2}\xi^{\sim 2}-a^{\sim 2}\eta^{\sim 2}\xi^{\sim 2}-a^{\sim 2}+a^{\sim 2}\eta^{\sim 2} \end{bmatrix}$$

Ezután kiszámolhatjuk a skálafaktorokat, és hozzáadhatjuk őket a rendszer azon belső táblázatához, amely a különböző koordinátarendszerekhez tartozó skálafaktorokat tárolja.

```
> h1 := sqrt( g[1,1] );
```

$$h1 := a^{\sim}\sqrt{\frac{\xi^{\sim 2}-\eta^{\sim 2}}{\xi^{\sim 2}-1}}$$

```
> h2 := sqrt( g[2,2] );
```

$$h2 := a^{\sim}\sqrt{\frac{\xi^{\sim 2}-\eta^{\sim 2}}{1-\eta^{\sim 2}}}$$

```
> h3 := sqrt( factor(g[3,3]) );
```

$$h3 := a^{\sim}\sqrt{(\xi^{\sim}-1)(\xi^{\sim}+1)(1-\eta^{\sim})(\eta^{\sim}+1)}$$

```
> sftable := readlib( 'linalg/scaletable' ):
> sftable[myprolatespheroidal] := subs(
> { a=_a, xi=_x, eta=_y, phi=_z }, [h1,h2,h3] );
```

$$sftable_{myprolatespheroidal} := [_a\sqrt{\frac{_x^2-_y^2}{_x^2-1}}, _a\sqrt{\frac{_x^2-_y^2}{1-_y^2}}, _a\sqrt{(_x-1)(_x+1)(1-_y)(_y+1)}]$$

A Maple ezeket a skálafaktorokat használja a gradiens, a Laplace-operátor stb. kiszámításánál.

```
> laplacian( f, c, coords=myprolatespheroidal(a) ):
> map( combine, " ):
> collect( ", [ diff(f,xi$2), diff(f,eta$2), diff(f,xi),
> diff(f,eta), f ], distributed, simplify );
```

$$-\frac{(\xi^{\sim 2}-1)\left(\frac{\partial^2}{\partial\xi^{\sim 2}}f\right)}{(-\xi^{\sim 2}+\eta^{\sim 2})a^2}+\frac{(-1+\eta^{\sim 2})\left(\frac{\partial^2}{\partial\eta^{\sim 2}}f\right)}{(-\xi^{\sim 2}+\eta^{\sim 2})a^2}-2\frac{\xi^{\sim}\left(\frac{\partial}{\partial\xi^{\sim}}f\right)}{(-\xi^{\sim 2}+\eta^{\sim 2})a^2}$$
$$+2\frac{\eta^{\sim}\left(\frac{\partial}{\partial\eta^{\sim}}f\right)}{(-\xi^{\sim 2}+\eta^{\sim 2})a^2}-\frac{\frac{\partial^2}{\partial\phi^{\sim 2}}f}{(-1+\eta^{\sim 2})a^2(\xi^{\sim}-1)(\xi^{\sim}+1)}$$

19.5. A Moore–Penrose inverz

A könyvben eddig általában nem úgy tekintettük a Maple-t, mint olyan programozási nyelvet, amellyel kiterjeszthetjük a rendszer beépített lehetőségeit. Néha azonban nagy segítség a programozhatóság. Az A mátrix A^+ *Moore–Penrose inverze* például [5] szerint a következő határértékből számítható ki:

$$A^+ = \lim_{x \to 0} \left(((A^T A + x^2 I)^{-1} A^T \right),$$

amennyiben ez a limesz létezik. A `linalg` csomag eszközeinek és a Maple-ről szerzett ismereteinknek a fölhasználásával képesek vagyunk rá, hogy kiterjesszük a rendszert.

```
> MPinv := A -> map( limit, evalm(
> evalm( (linalg[transpose](A)&*A+x^2)^(-1) )
> &* linalg[transpose](A) ), x=0 );
```

$$MPinv := A \to \mathrm{map}(limit,$$

$$\mathrm{evalm}(\mathrm{evalm}(\frac{1}{(linalg_{transpose}(A)\,`\&*`\,A) + x^2})\,`\&*`\,linalg_{transpose}(A))$$

$$, x = 0)$$

```
> M := randmatrix(3,2);
```

$$M := \begin{bmatrix} -85 & -55 \\ -37 & -35 \\ 97 & 50 \end{bmatrix}$$

```
> MPinv(M);
```

$$\begin{bmatrix} \frac{427}{88957} & \frac{2579}{88957} & \frac{2275}{88957} \\ \frac{-14093}{889570} & \frac{-45953}{889570} & \frac{-14939}{444785} \end{bmatrix}$$

```
> poly := proc() Randpoly(2,y) mod 2 end:
> M := randmatrix(3,2,entries=poly);
```

$$M := \begin{bmatrix} y^2 + y & y^2 + y + 1 \\ y^2 + y & y^2 \\ y^2 + y + 1 & y^2 + 1 \end{bmatrix}$$

```
> MPinv(M);
```

$$\left[-\frac{2\,y^5 + 4\,y^4 + 3\,y^3 + 3\,y^2 + y + 1}{\%1}, \frac{y\,(y^4 + 3\,y^3 + 5\,y^2 + 3\,y + 2)}{\%1}, \right.$$

$$\left. \frac{y^5 + 3\,y^4 + 3\,y^3 + 4\,y^2 + 2\,y + 1}{\%1} \right]$$

$$\left[\frac{2\,y^5+6\,y^4+7\,y^3+5\,y^2+2\,y+1}{\%1},\ -\frac{y\,(y^4+3\,y^3+4\,y^2+2\,y+1)}{\%1},\right.$$
$$\left.-\frac{y\,(y^4+3\,y^3+2\,y^2+y+1)}{\%1}\right]$$
$$\%1 := 2\,y+10\,y^3+7\,y^2+12\,y^4+1+2\,y^6+8\,y^5$$

19.6. Gyakorlatok

1. A PUMA robotkar [152] szerinti Denavit–Hartenberg paraméterei:

Csukló	α	θ	d	a
1	$-90°$	θ_1	0	0
2	$0°$	θ_2	0	a_2
3	$90°$	θ_3	d_3	a_3
4	$-90°$	θ_4	d_4	0
5	$90°$	θ_5	0	0
6	$0°$	θ_6	0	0

Számítsuk ki a robotkar csúcsának pozícióját és orientációját a kinematikai paraméterek függvényében, továbbá határozzuk meg a csúcs transzlációs és rotációs sebességét.

2. Az IRb-6 robotmanipulátor [180]-ban közölt paraméterei:

Csukló	α	θ	d	a
1	$90°$	θ_1	d_1	0
2	$0°$	θ_2	0	a_2
3	$0°$	θ_3	0	a_3
4	$90°$	θ_4	0	0
5	$0°$	θ_5	d_5	0

Számítsuk ki a robotkar csúcsának pozícióját és orientációját a kinematikai paraméterek függvényében, továbbá határozzuk meg a csúcs transzlációs és rotációs sebességét.

3. Tekintsük az emberi test kadmium-fölvételének a 19.4. ábrán látható négy szakaszos modelljét, melyet a következő egyenletek definiálnak:

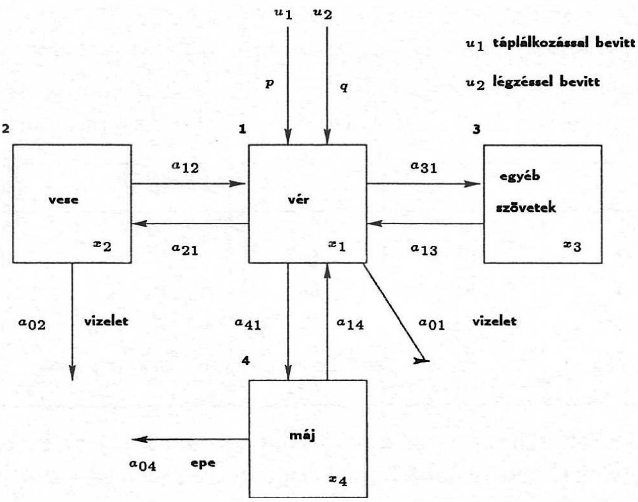
$$\begin{aligned}\dot{x}(t) &= A\,x(t) = B\,u(t), \qquad x(0) = x_0\\ y(t) &= C\,x(t),\end{aligned}$$

ahol

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14}\\ a_{21} & a_{22} & 0 & 0\\ a_{31} & 0 & a_{33} & 0\\ a_{41} & 0 & 0 & a_{44}\end{pmatrix}, \qquad B = \begin{pmatrix} p & q\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{pmatrix},$$

$$C = \begin{pmatrix} c_{11} & c_{12} & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & 0 & c_{44} \end{pmatrix},$$

továbbá $a_{11} = -(a_{01} + a_{21} + a_{31} + a_{41})$, $a_{22} = -(a_{02} + a_{12})$, $a_{33} = -a_{13}$, $a_{44} = -(a_{04} + a_{14})$, $c_{11} = a_{01}$ és $c_{12} = a_{02}$. Föltételezzük, hogy a p, q, c_{22} és a c_{44} paramétereket ismerjük.



19.4. ábra: A kadmium fölhalmozódása az emberi szervezetben

Használjuk fel a Maple-t annak bizonyítására, hogy ez a modell strukturálisan globálisan azonosítható.

4. Az ln-tan-henger koordinátákat a következő képletek definiálják:

$$\begin{aligned} x &= \frac{a}{\pi}\ln\left(\frac{\sinh^2\eta + \sin^2\psi}{\sinh^2\eta + \cos^2\psi}\right), \\ y &= \frac{2a}{\pi}\arctan\left(\frac{\sinh 2\eta}{\sin 2\psi}\right), \\ z &= z, \end{aligned}$$

ahol $0 < a$, $-1 \le \eta \le 1$ és $0 \le \psi \le 2\pi$. Határozzuk meg a koordinátarendszer skálafaktorait és fejezzük ki a Laplace-operátort ebben a koordinátarendszerben.

5. A molekula-pályák Hückel-elméletének fölhasználásával számítsuk ki a benzén π-elektron energiaszintjeit. Határozzuk meg a legkisebb energiájú állapot töltéseloszlását is.

Irodalomjegyzék

[1] A. ABRAMOWITZ – I. STEGUN: *Handbook of Mathematical Functions*, Dover Publishers, 1970.

[2] S.A. ABRAMOV – K.UY. KVASHENKO: *Fast Algorithms to Search for the Rational Solutions of Linear Differential Equations with Polinomial Coefficients* In: S. M. WATT (szerk.): Proceedings of ISSAC '91, pp. 267–270, ACM Press, 1991.

[3] J.M. AGUIREGABIRIA – A. HERNANDEZ – M. RIVAS: *Are We Careful Enough When Using Computer Algebra?*, Computers in Physics, 8: 56–61, 1994.

[4] A.G. AKRITAS: *Elements of Computer Algebra*, John Wiley & Sons, 1989.

[5] A.E. ALBERT: *Regression and the Moore–Penrose Pseudoinverse*, Academic Press, 1972.

[6] C.M. ANDERSEN – J.F. GEER: *Power Series Expansions for the Frequency and Period of the Limit Cycle of the van der Pool Equation*, SIAM J. Appl. Math., 42: 678–693, 1982.

[7] T. BANCHOFF: *Differential Geometry and Computer Graphics*, In: W. JÄGER – J. MOSER – R. REMMERT (szerk.): Perspectives in Mathematics, pp. 43–60. Birkhäuser Verlag, 1984.

[8] D. BARTON – J.P. FITCH: *Applications of Algebraic Manipulation Programs in Physics*, Rep. Prog. Phys., 35: 235–314, 1972.

[9] D. Barton – J.P. Fitch: *CAMAL: the Cambridge Algebra System*, SIGSAM Bull., 8(3): 17–23, 1974

[10] D. Barton – R. Zippel: *Polynomial Decomposition Algorithms*, J. Symbolic Computation, 1(2): 159–168, 1985.

[11] D. Barton – I.M. Willers – R.V.M. Zahar: *Taylor Series Methods for Ordinary Differential Equations – an Evaluation*, In: J.R. Rice (szerk.): Mathematical Software, pp. 369–389. Academic Press, 1972.

[12] C. Batut – D. Bernardi – H. Cohen – M. Oliver: *User's Guide to PARI-GP, version 1.39*, Université Bordeaux I, 1995.

[13] T. Becker – V. Weispfenning – H. Kredel: *Gröbner Bases*, Springer-Verlag, 1993.

[14] A. Berdnikov: *private communications*, RIACA, 1994.

[15] F. Bergeron: *Surprising Mathematics Using a Computer Algebra System*, J. Symbolic Computation, 15(3): 365–370, 1993.

[16] E.R. Berlekamp: *Factoring Polinomials over Large Finite Fields*, Math. Comp., 24: 713–715, 1970.

[17] W. Bosma – J. Cannon: *Handbook of Magma Functions*, Univ. of Sidney, 1994.

[18] W. Bosma – J. Cannon – C. Playoust – A. Steel: *Solving Problems with Magma*, Univ. of Sidney, 1994.

[19] A. Boyle – B.F. Caviness: *Future Directions for Research in Symbolic Computation*, SIAM Reports on Issues in the Mathematical Sciences, 1990.

[20] M. Bronstein – J.H. Davenport – B.M. Trager: *Symbolic Integration is Algorithmic*, Tutorial, Computers and Mathematics 1989, MIT, 1989.

[21] M. Bronstein: *Integration of Elementary Functions*, J. Symbolic Computation, 9(2): 117–174, 1990.

[22] M. Bronstein: Linear Ordinary Differential Equations: breaking through the order 2 barrier, In: P. Wang (szerk.): *Proceedings of ISSAC '92*, pp. 42–48. ACM Press, 1992.

[23] M. Bronstein – B. Salvy: *Full Partial Fraction Decomposition of Rational Functions*, In: M. Bronstein (szerk.): Proceedings of ISSAC '93, pp. 157–160. ACM Press, 1993.

[24] D.A. Brown: *Quantum Chemistry*, Penguin Books, 1972.

[25] C. de Bruijn: *De structurele identificeerbaarheid en het schatbaar zijn van de modelparameters van het compartimentele model foor de verspreiding van cadmium in het menselijk lichaam*, Technical Report, RIVM (in Dutch), 1990.

[26] B. BUCHBERGER: *Gröbner Bases: An Algorithmic Method in Polinomial Ideal Theory*, In: N.K. BOSE (szerk.): Progress, Directions and Open Problems in Multidimensional Systems Theory, pp. 184–232. Reidel Publishing Company, 1985.

[27] B. BUCHBERGER: *Applications of Gröbner Bases in Non-linear Computational Geometry*, In: J.R. RICE (szerk.): Mathematical Aspects of Scientific Software, IMA Volume in Mathematics and is Applications 14, pp. 59–87. Springer-Verlag, 1988.

[28] G. BUTLER – J. CANNON: *The Design of Cayley – A Language for Modern Algebra*, In: A. MIOLA (szerk.): Design and Implementation of Symbolic Computation Systems, Lecture Notes in Computer Science 429, pp. 10–19. Springer-Verlag, 1990.

[29] P.F. BYRD – M.D. FRIEDMAN: *Handbook of Elliptic Integrals for Engineers and Physicists*, volume LXVII of Die Grundlagen der Mathematischen Wissenschaften, Springer-Verlag, 1971.

[30] J. CALMET – J.A. VAN HULZEN: *Computer Algebra Applications*, In: B. BUCHBERGER – G.E. COLLINS – R. LOOS, (szerk.): Computer Algebra – Symbolic and Algebraic Computation, pp. 245–258. Springer-Verlag, 1983.

[31] J. CANNON – W. BOSMA: *Cayley Quick Reference Guide*, Univ. of Sydney, 1991.

[32] J. CANNON – J. PLAYOUST: *An Introduction to Magma*, Univ. of Sydney, 1994.

[33] D.G. CANTOR – H. ZASSENHAUS: *A New Algorithm over Large Finite Fields*, Math. Comp., 36: 587–592, 1981.

[34] A. CAPANI – G. NIESI: *CoCoa User's Manual*, Univ. of Genova, CoCoa version 3.0b, 1995.

[35] J. CARMINATI – J S. DEVITT – G. J. FEE: *Isogroup of Differential Equations Using Algebraic Computing*, J. Symbolic Computation, 14(1): 103–120, 1992.

[36] B.W. CHAR – K.O. GEDDES – W.M. GENTLEMAN – G.H. GONNET: *The Design of Maple: A Compact, Portable, and Powerful Computer Algebra System*, In: J.A. VAN HULZEN (szerk.): Computer Algebra (Proceedings of EUROCAL '83), Lecture Notes in Computer Science 162, pp. 101–115, Springer-Verlag, 1983.

[37] B.W. CHAR – K.O. GEDDES – G.H. GONNET: *GCDHEU: Heuristic GCD Algorithm Based on Integer GCD Computation*, J. Symbolic Computation, 7(1): 31–48, 1989.

[38] B.W. CHAR – K.O. GEDDES – G.H. GONNET – B.L. LEONG – M.B. MONAGAN – S.M. WATT: *Maple V Library Reference Manual*, Springer-Verlag, first edition, 1991.

[39] B.W. Char – K.O. Geddes – G.H. Gonnet – B.L. Leong – M.B. Monagan – S.M. Watt: *Maple V Language Reference Manual*, Springer-Verlag, first edition, 1991.

[40] B.W. Char – K.O. Geddes – G.H. Gonnet – B.L. Leong – M.B. Monagan – S.M. Watt: *First Leaves: A Tutorial Introduction*, Springer-Verlag, first edition, 1992.

[41] P.L. Chebyshev: *Sur l'integration des differentielles qui contienent une racine carrée d'un polinôme du troisième ou du quatrième degré*, Oeuvres de P. L. Tchebychef, volume I, pp. 171–200. Chelsea, 1957.

[42] S.M. Christensen: *Resources for Computer algebra*, Computers in Physics, 8: 308–315, 1994.

[43] A.M. Cohen: *Computer algebra, Theory and Practice*, Nieuw Arch. voor Wisk., IV(7): 215–230, 1989.

[44] A.M. Cohen – G.C.M. Ruitenburg: *Generating Functions and Lie Groups*, In: A.M. Cohen (szerk.): Computational Aspects of Lie Group Representations and Related Topics, CWI Tract 84, pp. 19–28. CWI, 1991.

[45] A.M. Cohen – R.L. Griess Jr. – B. Lisser: *The Group* $L(2,61)$ *embeds in the Lie Group of Type* E_8, Comm. Algebra, 21: 1889–1907, 1993.

[46] A.M. Cohen – J.H. Davenport – A.J.P. Heck: *An Overview of Computer Algebra*, Computer Algebra for Industry: Problem Solving in Practice, pp. 1–52. John Wiley & Sons, 1993.

[47] J.W. Cooley – J.W. Tukey: *An Algorithm for the Machine Calculation of Complex Fourier Series*, Math. Comp., 19: 297–301, 1965.

[48] R.M. Corless – G.H. Gonnet – D.E.G. Hare – D.J. Jeffrey: *On Lambert's W Function*, Preprint in Maple Share Library, 1993.

[49] R.M. Corless: *Simplification and the Assume Facility*, Maple Technical Newsletter, 1(1): 24–31, 1994.

[50] R.M. Corless – K. El–Sawy: *Solution of Banded Linear Systems in Maple Using LU Factorization*, In: R.J. Lopez (szerk.): Maple V: Mathematics and its Applications, pp. 219–227. Birkhäuser Verlag, 1994.

[51] D. Cox – J. Little – D. O'Shea: *Ideals, Varieties, and Algorithms*, Springer-Verlag, 1992.

[52] J.H. Davenport: *On the Integration of Algebraic Functions*, Lecture Notes in Computer Science 102, Springer-Verlag 1981.

[53] J.H. Davenport – Y. Siret – E. Tournier: *Computer algebra: systems and algorithms for algebraic computation*, Academic Press, 1988.

[54] J.H. DAVENPORT – B.M. TRAGER: *Scratchpad's View of Algebra I: Basic Commutative Algebra*, In: A. MIOLA (szerk.): Design and Implementation of Symbolic Computation Systems, Lecture Notes in Computer Science 429, pp. 40–54. Springer-Verlag, 1990.

[55] J.H. DAVENPORT: *The Axiom System*, Proceedings of NAGUA '91, NAG Ltd., 1991.

[56] J. DIEUDONNÉ: *Treatise on Analysis*, volume 10–III of Pure and Applied Mathematics, Academic Press, 1972.

[57] A. DINGLE – R. FATEMAN: *Branch Cuts in Computer Algebra*, Proceedings of ISSAC '94, pp. 250–257. ACM Press, 1994.

[58] W.H. ENRIGHT: *The Relative Efficiency of Alternative Defect Control Schemes for High Order Continuous Runge–Kutta Formulas*, Technical Report 252/91, University of Toronto, Dept. of Computer Science, June 1991.

[59] A.V.D. ESSEN: *Polinomial Maps and the Jacobian Conjecture*, Computational Aspects of Lie Group Representations and Related Topics, CWI Tract 84, pp. 29–44. CWI, 1991.

[60] R.J. FATEMAN: *Advances and Trends in the Design and Construction of Algebraic Manipulation Systems*, In: S. WATANABE – M. NAGATA (szerk.): Proceedings of ISSAC '90, pp. 60–67. ACM Press, 1990.

[61] E. FEHLBERG: *Klassische Runge–Kutta Formeln vierter und niedriger Ordnung mit Schrittweiten Kontrolle und ihre Anwendung in Wärmeleitungsprobleme*, Computing, 6: 61–71, 1970.

[62] R.P. FEYNMAN – R.B. LEIGHTON – M. SANDS: *The Feynman Lectures on Physics*, volume II. Addison–Wesley, 1964. fourth printing, pp. 22–8.

[63] J. FITCH: *Solving Algebraic Problems with REDUCE*, J. Symbolic Computation, 1(2): 211–228, 1985.

[64] I. FOSTER – S. TAYLOR: *Strand – New Concepts in Parallel Progamming*, Prentice-Hall, 1989.

[65] I. FRICK: *SHEEP User's Manual*, Univ. of Stockholm, 1977.

[66] F.N. FRITSCH – R.E. SHAFER – W.P. CROWLEY: *Solution of the Transcendental Equation* $we^w = \chi$, Comm. ACM., 16: 123–124, 1973.

[67] FUCHSTEINER ET AL.: *MuPAD: Multi Processing Algebra Data Tool; Benutzerhandbuch; MuPAD Version 1.1*, Birkhäuser Verlag, 1993

[68] FUCHSTEINER ET AL.: *MuPAD: Multi Processing Algebra Data Tool; Tutorial*, Birkhäuser Verlag, 1993

[69] W. GANDER – J. HREBÍCEK – S. BARTON: *The Tractrix and Similar Curves*, In: W. GANDER – J. HREBÍCEK (szerk.): Solving Problems in Scientific Computing Using Maple and Matlab, pp 1–14. Springer-Verlag, 1995.

[70] C.W. GEAR: *Numerical Initial Value Problems in Ordinary Differenctial Equations*, Prentice-Hall, 1971.

[71] K.O. GEDDES: *Numerical Integration in a Symbolic Context*, In: B. CHAR (szerk.): Proceedings of SYMSAC '86, pp 185–191. ACM Press, 1986.

[72] K.O. GEDDES – G.H. GONNET: *A New Algorithm for Computing Symbolic Limits Using Hierarchical Series*, In: P. GIANNI (szerk.): Symbolic and Algebraic Computation, Proceedings ISSAC '88, Lecture Notes in Computer Science 358, pp 490–495. Springer-Verlag, 1989.

[73] K.O. GEDDES – T.C. SCOTT: *Recipes for Classes of Definite Integrals Involving Exponentials and Logarithms*, In: E. KALTOFEN – S.M. WATT (szerk.): Computers and Mathematics 1989, pp. 192–201. Academic Press, 1989.

[74] K.O. GEDDES – L.Y. STEFANUS: *On the Risch–Norman integration method and its implementation in Maple*, In: G.H. GONNET (szerk.): Proceedings of ISSAC '89, pp. 212–217. ACM Press, 1989.

[75] K.O. GEDDES: *Numerical Integration using Symbolic Analysis*, Maple Technical Newsletter, 6: 8–17, 1991.

[76] K.O. GEDDES – S.R. CZAPOR – G. LABAHN: *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1992.

[77] K.O. GEDDES: *A Package for Numerical Approximation*, Maple Technical Newsletter, 10: 28–36, 1993.

[78] A. GIOVINI – G. NIESI: *CoCoA: A User-Friendly System for Commutative Algebra*, In: A. MIOLA (szerk.): Design and Implementation of Symbolic Computation Systems, Lecture Notes in Computer Science 429, pp. 20–29. Springer-Verlag, 1990.

[79] G.H. GONNET – D.W. GRUNTZ: *Algebraic Manipulation: Systems*, In: A. RALSTON ET AL. (szerk.): Encyclopedia of Computer Science Engineering, Van Nostrand Reinhold, 1991.

[80] G.H. GONNET – D.W. GRUNTZ: *Limit Computation in Computer Algebra*, Technical Report 187, Department of Computer Science, ETH Zürich, 1992.

[81] M.J. GONZÁLEZ-LÓPEZ – T. RECIO: *The ROMIN inverse geometric model and the dynamic evaluation method*, In: A.M. COHEN (szerk.): Computer Algebra for Industry: Problem Solving in Practice, pp. 117–141. John Wiley & Sons. 1993.

[82] R. W. GOSPER: *Decision Procedure for Indefinite Hypergeometric Summation*, Proc. Natl. Acad. Sci. USA, 75: 40–42, 1978.

[83] X. GOURDON – B. SALVY: *Computing One Million Digits of π*, Maple Technical Newsletter, 10: 66–71, 1993.

[84] I. S. GRADSHTEYN – I. M. RYZHIK: *Table of Integrals, Series and Products*, Academic Press, fifth edition, 1994.

[85] J. GRAF V. SCHMETTOW: *KANT – a Tool for Computations in Algebraic Number Fields*, In: A. PETHŐ – M. POHST – H. WILLIAMS – H. ZIMMER (szerk.): Computational Number Theory, pp. 321–330. de Gruyter, 1991.

[86] D.R. GRAYSON: *Macaulay 2*,
`http://www.math.uiuc.edu/~dan/Macaulay2/`, 1995.

[87] A. GRIEWANK: *On Automatic Differentiation*, In: M. IRI – K. TANABE (szerk.): Mathematical Programming, pp. 83–107. Kluwer Academic Publishers, 1989.

[88] A. GRIEWANK: *The Chain Rule Revisited*, Scientific Computing. SIAM News, pp. 20–21 (part I), 8–9, 24 (part II), May (part I), July (part II) 1991.

[89] A. GRIEWANK – G.F. CORLISS: *Automatic Differentation of Algorithms: Theory, Implementation and Application*, Proceedings in Applied Mathematics 53., SIAM, Philadelphia, 1991.

[90] E. GROSWALD: *Bessel Polynomials*, Lecture Notes in Mathematics 698., Springer-Verlag, 1980.

[91] J. GROTENDORST: *A Maple Package for Transforming Series, Sequences, and Functions*, Comp. Phys. Comm., 67: 325–342, 1991.

[92] J. GUTIÉRREZ – T. RECIO – C. RUIZ DE VELASCO: *Polynomial decomposition algorithm of almost quadratic complexity*, In: T. MORA (szerk.): Proceedings of AAECC-6, Lecture Notes in Computer Science 357., Springer-Verlag, 1989.

[93] J. GUTIÉRREZ – T. RECIO: *A Practical Implementation of Two Rational Decomposition Algorithms*, In: P. WANG (szerk.): Proceedings of ISSAC '92, pp. 152-157. ACM Press, 1992.

[94] J. GUTIÉRREZ – T. RECIO: *Rational Function Decomposition and Gröbner Basis in the Parametrization of Plane Curves*, Proceedings of LATIN '92. Sao Paulo, Brazil, Springer Lecture Notes of Comput. Sci. 583. pp. 239-245. Springer-Verlag, 1992.

[95] D. HARPER – C. WOOFF – D. HODGKINSON: *A Guide to Computer Algebra Systems*, John Wiley & Sons, 1991.

[96] B.K. HARRISON – F.B. ESTABROOK: *Geometric Approach to Invariance Groups and Solutions of Partial Differential Equations*, J. Math. Phys., 12: 653–665, 1971.

[97] A.C. HEARN: *REDUCE User's Manual.*, The Rand Corporation, Santa Monica, California, 1987.

[98] A.J.P. HECK: *Transformation between Geocentric and Geodetic Coordinates*, In: A.M. COHEN (szerk.): Computer Algebra for Industry: Problem Solving in Practice, pp. 203–219. John Wiley & Sons, 1993.

[99] A.J.P. HECK: *FORM for Pedestrians*, `http//www.can.nl/ SystemsOverview/General/FORM/`, CAN Expertise Center & Univ. of Petrópolis, 1993.

[100] A.J.P. HECK: *Computer Algebra: A Tool in Identifiability Testing*, In: A.M. COHEN – L. VAN GASTEL – S. VERDUYN LUNEL (szerk.): Computer Algebra for Industry 2: Problem Solving in Practice, pp. 267–289. John Wiley & Sons, 1995.

[101] W. HEREMAN: *Symbolic Software for Lie Symmetry Analysis*, In: N.H. IBRAGIMOV (szerk.): CRC Handbook of Lie Group Analysis of Differential Equations, Vol.: New Trends in Theoretical Developments and Computational Methods, chap. 13, CRC Press, 1995.

[102] A.C. HINDMARSH: *ODEPACK: a Systemized Collection of ODE Solvers*, In: R. STEPLEMAN (szerk.): Numerical Methods for Scientific Computation, North-Holland, 1983.

[103] C. HOLLINGER – P. SERF: *SIMATH – a Computer Algebra System*, In: A. PETHO – M. POHST – H. WILLIAMS – H. ZIMMER, (szerk.): Computational Number Theory, pp. 331-342. de Gruyter, 1991.

[104] L. HORNFELDT: *STENSOR Reference Manual*, Univ. of Stockholm, 1988.

[105] E. HOROWITZ: *Algorithms for Partial Fraction Decomposition and Rational Integration*, In: S.R. PETRICK (szerk.): Proceedings of SYM-SAM '71, pp. 441–457. ACM Press, 1971.

[106] J.A. VAN HULZEN – J. CALMET: *Computer Algebra Systems*, In: B. BUCHBERGER – G.E. COLLINS – R. LOOS (szerk.): Computer Algebra – Symbolic and Algebraic Computation, pp. 221-244. Springer-Verlag, 1983.

[107] A. IVIC: *The Riemann Zeta Function*, John Wiley & Sons, 1985.

[108] R.D. JENKS – R.S. SUTOR – S.M. WATT: *Scratchpad II: And Abstract Datatype System for Mathematical Computing*, In: J.R. RICE (szerk.): Mathematical Aspects of Scientific Software, IMA Volume in Mathematics and its Applications 14, pp. 157–182. Springer-Verlag, 1988.

[109] R.D. JENKS – R.S. SUTOR: *AXIOM, The Scientific Computation System*, Springer-Verlag, 1992.

[110] E.W. JOHNSON: *Linear Algebra with Maple V*, Brooks/Cole, 1993.

[111] W. KAHAN: *Branch Cuts for Complex Elementary Functions or Much Ado About Nothing's Sign Bit*, In: A. ISERLES – M.J.D. POWELL (szerk.): The State of the Art in Numerical Analysis, pp. 65–212. Clarendon Press, 1987.

[112] E. KALTOFEN: *Polynomial Factorization*, In: B. BUCHBERGER – R. LOOS (szerk.): Computer Algebra – Symbolic and Algebraic Computation, pp. 95–114. Springer-Verlag, 1983.

[113] E. KALTOFEN: *Sparse Hensel Lifting*, In: B.F. CAVINESS (szerk.): Proceedings of EUROCAL '85, Lecture Notes in Computer Science 204, Vol. 2, pp. 4–17. Springer-Verlag, 1985.

[114] E. KALTOFEN: *Polynomial Factorization 1982–1986*, In: D.V. CHUDNOVSKY – R.D. JENKS (szerk.): Computers & Mathematics, Lecture Notes in Pure and Applied Mathematics 125, pp. 285–309. Marcel Dekker, 1990.

[115] E. KALTOFEN: *Polynomial Factorization 1987–1991*, Proc. of LATIN '92, Sao Paulo, Brazil, Springer Lecture Notes of Comput. Sci. 583. pp. 294–313. Springer-Verlag, 1992.

[116] M. KLERER – F. GROSSMAN: *Error Rates in Tables of Indefinite Integrals. Industrial Mathematics*, 18, 1968.

[117] D.E. KNUTH: *The Art of Computer Programming, Vol. II, Seminumerical Algorithms*, Addison–Wesley, second edition, 1981.

[118] D.E. KNUTH: *Problem E3335*, Amer. Math. Monthly, 96: 525, 1989.

[119] H.-P. KO: *Geometry Theorem Proving by Decomposition of Quasi-Algebraic Sets: An Application of the Ritt–Wu Principle*, In: D. KAPUR – J.L. MUNDY (szerk.): Geometrical Reasoning, pp. 95–122. MIT Press, 1989.

[120] W. KOEPF: *Algorithms for the Indefinite and Definite Summation*, Technical Report, Preprint SC 94-33, Konrad–Zuse–Zentrum Berlin (ZIB). December 94.

[121] J. KOVACIC: *An Algorithm for Solving Second Order Homogeneous Differential Equations*, J. Symbolic Computation, 2(1): 3–43, 1986.

[122] D. KRAFT: *Modeling and Simulating Robots in Maple*, Maple Technical Newsletter, 1(2): 39–47, 1994.

[123] G. LABAHN: *Solving Linear Differential Equations in Maple*, Maple Technical Newsletter, 2(1): 20–28, 1995.

[124] L. LAMPORT: *LaTeX, A Document Preparation System*, Addison–Wesley, 1988.

[125] D. LAZARD – R. RIOBOO: *Integration of Rational Functions: Rational Computation of the Logarithmic Part* J. Symbolic Computation, 9(2): 113–116, 1990.

[126] Y. LECOURTIER – A. RAKSANYI: *The Testing of Structural Properties Through Symbolic Computation*, In: E. WALTER (szerk.): Identifiabilty of Parametric Models, Pergamon Press, 1987.

[127] M.A. VAN LEEUWEN, A.M. COHEN, – B. LISSER: *LiE: A Package for Lie Group Computations*, CAN Expertise Center, 1992.

[128] A. LEHTONEN: *The Klein Bottle. The Mathematica Journal*, 1(3): 65, 1991.

[129] A.K. LENSTRA: *Factoring Polynomials over Algebraic Number Fields*, In: J.A. VAN HULZEN (szerk.): Computer Algebra (Proceedings of EU-ROCAL '83), Lecture Notes in Computer Science 162, pp. 245–254. Springer-Verlag, 1983.

[130] A.H.M. LEVELT: *Various Problems Solved by Computer Algebra*, In: A.M. COHEN – L. VAN GASTEL – S. VERDUYN LUNEL (szerk.): Computer Algebra for Industry 2: Problem Solving in Practice, pp. 43–59. John Wiley & Sons, 1995.

[131] B. LISSER: *Konstant's Conjecture*, Maple Technical Newsletter, Special Issue: „Maple in Mathematics and the Sciences." pp. 29–34, 1994.

[132] R. LOOS: *Computing in Algebraic Extensions*, In: B. BUCHBERGER – G.E. COLLINS – R. LOOS (szerk.): Computer Algebra – Symbolic and Algebraic Computation, pp. 173–187. Springer-Verlag, 1983.

[133] J.V.D. LUNE – H.J.J.TE RIELE, – D.T. WINTER: *On the zeros of the Riemann zeta function*, Math. Comp., 46: 667–681, 1986.

[134] M. MACCALLUM – J. SKEA: *SHEEP, a computer algebra system for general relativity*, In: M. MACCALLUM – J. SKEA – J. MCCREA – R. MCLENAGHAN (szerk.): Algebraic Computing in General Relativity, pp. 1–172. Clarendon Press, 1994.

[135] M.A.H. MACCALLUM: *Using Computer Algebra to Solve Ordinary Differential Equations*, In: A.M. COHEN – L. VAN GASTEL – S. VERDUYN LUNEL (szerk.): Computer Algebra for Industry 2: Problem Solving in Practice, pp. 19–41. John Wiley & Sons, 1995.

[136] *MACSYMA User's Guide*, MACSYMA User's Guiden, System Reference Manual, and Mathematics Reference Manual, Macsyma, Inc., 1992.

[137] MATHSOURCE: `http://www.wri.com/mathsource`

[138] M. MIGNOTTE: *Mathematics for Computer Algebra*, Springer-Verlag, 1992.

[139] B. MISHRA: *Algorithmic Algebra*, Springer-Verlag, 1993.

[140] R. MOENCK: *On Computing Closed Forms for Summation*, Proc. MACSYMA User's Conf., pp. 225–236, 1977.

[141] M.B. MONAGAN: *Tips for Maple Users*, Maple Technical Newsletter, 1(2): 11–13, 1994.

[142] P. MOON – D.E. SPENCER: *Field Theory Handbook: Including Coordinate Systems, Differential Equations and Their Solutions*, Springer-Verlag, 1961.

[143] J. MOSES: *Symbolic Integration: The Stormy Decade*, Comm. ACM., 14: 548–560, 1971.

[144] A.H. NAYFEH – D.T. MOOK: *Nonlinear Oscillations*, John Wiley & Sons, 1979.

[145] B. NOBLE – M.A. HUSSAIN: *Multiple Scaling and a Related Expansion Method, with Applications*, Report BICOM 87/7, Brunel Univ., Uxbridge, England, June 1987.

[146] V.W. NOONBURG: *A Neural Network Modeled by an Adaptive Lotka–Volterra System*, SIAM J. Appl. Math, 49: 1779-1792, 1989.

[147] A.M. ODLYZKO: *Analytic Computations in Number Theory*, In: W. GAUTSCHI (szerk.): Mathematics of Computation 1943–1993: a Half-Century of Computational Mathematics, pp. 451–463. Proceedings of Symposia in Applied Mathematics, American Mathematical Society, 1994.

[148] G.J. OLDENBORGH: *An Introduction to FORM*, Univ. of Leiden, `http://www.can.nl/SystemsOverview/General/FORM/`

[149] P.J. OLVER: *Applications of Lie Groups to Differential Equations*, Springer-Verlag, 1986.

[150] M.K. PAUL: *A Note on Computation of Geodetic Coordinates from Geocentric (Cartesian) Coordinates*, Bull. Géodésique, 108: 135–139, 1973.

[151] R.P. PAUL: *Robot Manipulators: Mathematics, Programming and Control* MIT Press, 1981.

[152] R.P. PAUL: *Kinematic Control Equations for Simple Manipulators*, In: C. LEE – R. GONZALEZ – K. FU (szerk.): Tutorial on Robotics, pp. 66–72. IEEE Computer Society Press, 1983.

[153] P. PAULE – V. STREHL: *Symbolic Summation – Some Recent Developments*, Technical Report 95–11, RISC-Linz, Johannes Kepler University, Linz, Austria, 1995. (lásd még: J. FLEISCHER – J. GRABMEIER – F. HEHL – W. KÜCHLIN (szerk): Computer Algebra in Science and Engineering – Algorithms, Systems, and Applications, World Scientific, Singapore. In preparation.)

[154] R. PAVELLE: *Problems sent to the USENET sci.math.symbolic bulletin*

board, Archived in the REDUCE network library, 1989.

[155] P.P. Petrushev – V.A. Popov: *Rational Approximation of Real Functions*, Cambridge University Press, 1987.

[156] M.E. Pohst: *Computational Algebraic Number Theory*, volume 21 of DMV Seminar, Birkhauser Verlag, 1993.

[157] A. Raksanyi, Y. Lecourtier, E. Walter, – A. Venot: *Identifiability and Distinguishability Testing Via Computer Algebra*, Math. Biosciences, 77: 245–266, 1985.

[158] R.H. Rand: *Computer Algebra in Applied Mathematics: An Introduction to MACSYMA*, Research Notes in Mathematics 94, Pitman Publishing, 1984.

[159] R.H. Rand – D. Armbruster: *Perturbation Metods, Bifurcation Theory and Computer Algebra*, Applied Mathematical Sciences 65 Springer-Verlag, 1987.

[160] D. Redfern: *The Maple Handbook, Maple V Release 4*, Springer-Verlag, 1995.

[161] E.Ya. Remez: *Sur un procédé convergent d'approximation succesives pour déterminer les polynomes d'approximation*, Comptes Rendues, 193: 2063–2065, 1934.

[162] E.Ya. Remez: *Sur le calcul effectif des polynomes d'approximation de Tschebyscheff*, Comptes Rendues, 199: 337–340, 1934.

[163] R.H. Risch: *The problem of integration in finite terms*, Trans. AMS, 139: 167–189, 1969.

[164] M. Rothstein: *Aspects of Symbolic Integration and Simplification of Exponential and Primitive Functions*, PhD thesis, Univ. of Wisconsin, Madison, 1976.

[165] M. Schoenert et al: *GAP: Groups, Algorithms and Programming*, RWTH Aachen, 1994. GAP Manual Release 3.4,
`http://www.math.rwth/aachen.de:800/GAP/MANUAL/`

[166] F.W. Schwarz: *Symmetries of Differential Equations: From Sophus Lie to Computer Algebra*, SIAM Review, 30: 450–481, 1988.

[167] T.C. Scott – Y.B. Band, – K.O. Geddes: *Recipes for Solving Broad Classes of Definite Integrals and Applications*, Maple Technical Newsletter, 10: 19–27, 1993.

[168] K. Siegl: *Parallelizing Algorithms for Symbolic Computation Using MAPLE*, Technical Report 93-08, RISC-Linz, Johannes Kepler University, Linz, Austria, 1993. Published in: 4th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming, San Diego, CA, May 19–21, 1993.

[169] T.J. SMEDLEY: *Fast Methods for Computation with Algebraic Numbers*, PhD thesis, Univ. of Waterloo, 1990.

[170] B.K. SPEARMAN – K.S. WILLIAMS: *Characterization of Solvable Quintics* $x^5 + ax + b$, Amer. Math. Monthly, 101: 986–992, 1994.

[171] M.R. SPIEGEL: *Mathematical Handbook of Formulas and Tables*, McGraw-Hill, 1968.

[172] H. STEPHANI: *Differential Equations: Their Solution Using symmetries*, Cambridge University Press, 1993.

[173] M. STILLMAN, M. STILLMAN – D. BAYER: *Macaulay User Manual*, 1989.

[174] J. STOER – R. BULIRSCH: *Introduction to Numerical Analysis. Text in Applied Mathematics 12.* Springer-Verlag, second edition, 1993.

[175] D. STOUTEMYER: *Crimes and Misdemeanors in the Computer Algebra Trade*, Notices of the AMS, 38: 778–785, 1991.

[176] D. STOUTEMYER: *Derive User Manual*, Soft Warehouse, Inc., Honolulu, Hawaii, 1994. seventh edition.

[177] V. STREHL: *Binomial Sums and Identities*, Maple Technical Newsletter, 10: 37–49, 1993.

[178] H. STRUBBE: *Manual for SCHOONSCHIP*, Comput. Phys. Commun., 8: 1–30, 1974.

[179] R.S. SUTOR (SZERK.): *The Scratchpad II Computer Algebra System Interactive Environment Users Guide* Technical Report, IBM Thomas J. Watson Research Center, Yorktown Heights, 1988.

[180] T. SCKODNY: Modelling of Kinematics of the Irb-6 Manipulator. *Computers Math. Applic.*, 29: 77–94, 1995.

[181] R.G. TOBEY: *Algorithms for Antidifferentiation of Rational Functions* PhD thesis, Univ. of Wisconsin, Madison, 1967.

[182] B. TRAGER: *Algebraic Factoring and Rational Function Integration*, In: R.D. JENKS (szerk.): Proceedings of SYMSAC '76, pp. 219–226. ACM Press, 1976.

[183] B. TRAGER: *Integration of Algebraic Functions*, PhD thesis, MIT, 1984.

[184] J.A.M. VERMASEREN: *Symbolic Manipulation with FORM*, CAN Expertise Center, 1991.

[185] J.A.M. VERMASEREN: *Symbolic Heroics*, CAN Newletter, 11: 57–58, 1993.

[186] D. WANG: *An Implementation of the Characteristic Set Method in Maple*, Technical Report, 91-25, RISC-Linz, 1991.

[187] WATERLOO MAPLE INC.: *The Maple Learning Guide*, 1995.

[188] WATERLOO MAPLE INC.: *The Maple Programming Guide*, 1995.

[189] T. WEIBEL – G.H. GONNET: *An Algebra of Properties*, Proceedings of ISSAC '91, pp. 352–359. ACM Press, 1991.

[190] T. WEIBEL – G.H. GONNET: *An Algebra of Properties.* Technical Report 158, Informatik ETH-Zürich, 1991.

[191] T. WEIBEL – G.H. GONNET: *An Assume Facility for CAS, with a Sample Implementation for Maple*, In J. FITCS (szerk.): Design and Implementation of Symbolic Computation Systems, Lecture Notes in Computer Science 721, pp. 95–103. Springer-Verlag, 1992.

[192] E.J. WENIGER: *Nonlinear Sequence Transformations for the Acceleration of Convergence and the Summation of Divergent Series*, Comp. Phys. Reports, 10: 189–371, 1989.

[193] H.S. WILF – D. ZEILBERGER: *Rational functions certify combinatorial identities*, J. Amer. Math. Soc., 3: 147–158, 1990.

[194] H.S. WILF – D. ZEILBERGER: *Towards computerized proofs of identities*, Bull. of the Amer. Math. Soc., 23: 77–83, 1990.

[195] F. WINKLER: *Computer Algebra: Problems and Developments*, In: A.M. COHEN – L. VAN GASTEL – S. VERDUYN LUNEL (szerk.): Computer Algebra for Industry 2: Problem Solving in Practice, pp. 1–18 John Wiley & Sons, 1995.

[197] C. WOOFF – D. HODGKINSON: *MuMath: A Microcomputer Algebra System*, Academic Press, 1987.

[198] P.E.S. WORMER – F. DE GROOT: *The Potential Energy Surface of Triplet: A Representation in Hyperspherical Coordinates*, J. Chem. Phys. 1989: 2344, 90.

[199] *Számítógépes algebrai WWW szerverek:*
CAIN, `www.can.nl`
Computer Algebra Fachgruppe, `www.uni-karlsruhe.de~CAIS`
GDR MEDICIS, `medicis.polytechnique.fr`
RISC-Linz, `info.risc.uni-linz.ac.at`
SymbolicNet, `symbolicnet.mcs.kent.edu`.

[200] S.Y. YAN: *Primality Testing of Large Numbers in Maple*, Computers Math. Applic., 29: 1–8, 1995.

[201] R. ZIPPEL: *Rational Function Decomposition*, In: S.M. WATT (szerk.): Proceedings of ISSAC '91, pp. 1–6. ACM Press, 1991.

[202] R. ZIPPEL: *Effective Polynomial Computation*, Kluwer Academic Publishers, 1993.

Tárgymutató

Kiadja: JGYF Kiadó
Felelős kiadó: Pitrik József menedzserigazgató
Terjedelem: 42,75 ív (B5)

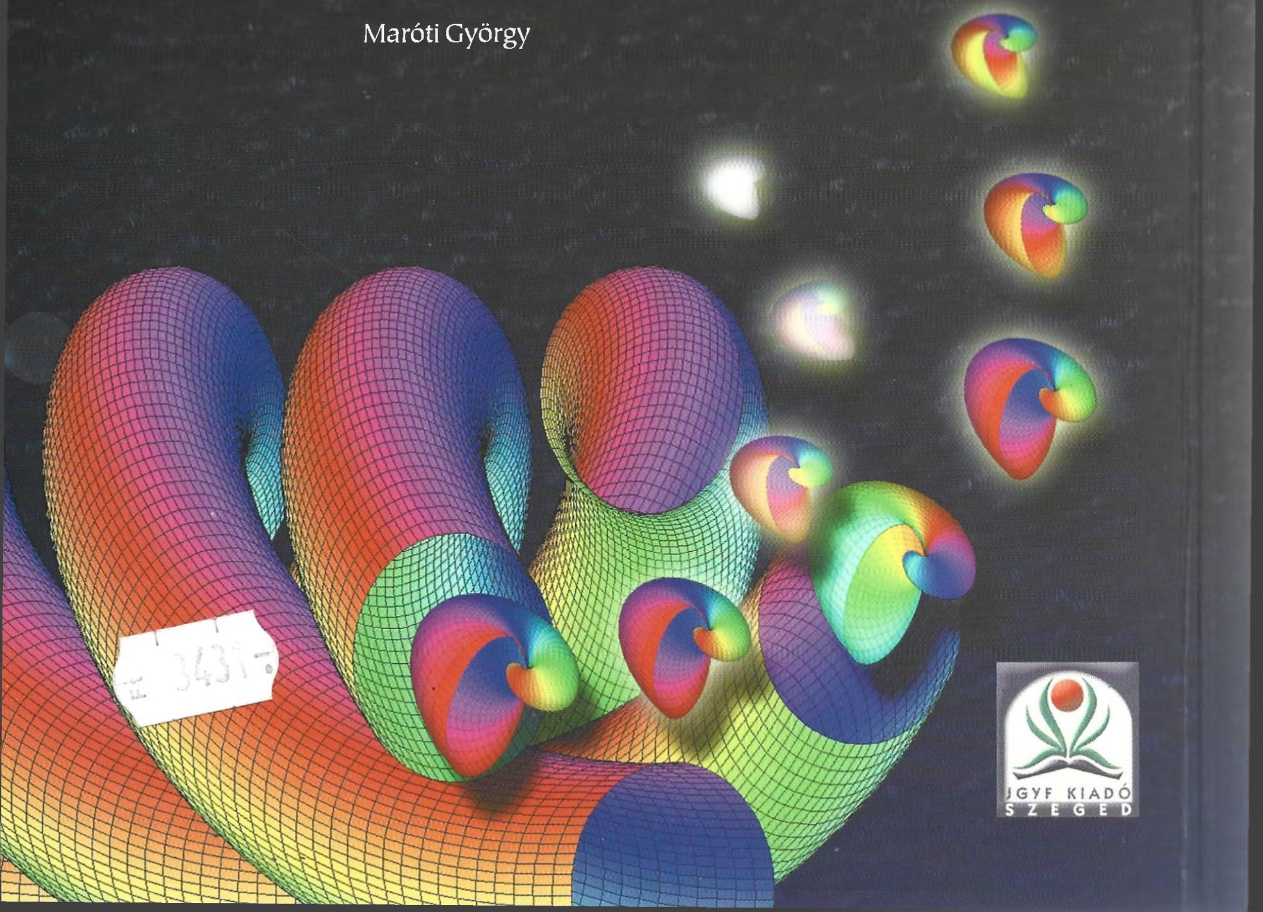
Nyomdai és kötészeti munka:
Letter-Print Kft., Budapest

Nem könnyű a Maple-ről jó könyvet írni. A Maple mindenekelőtt szoftver termék, s mint ilyet meg kell tanulni kezelni. Így azután nagy a kísértés felhasználói kézikönyv megírására. Igen ám, de a Maple kitűnő segédeszköz is matematikai, mérnöki problémák megoldására, csak éppen el kell sajátítani hozzá egy speciális gondolkodásmódot. Végül a Maple szemléltető eszköz a jó előadó kezében, valamint a tapasztalatgyűjtés, a kísérletezés eszköze az érdeklődő diák számára.

André Heck könyvének középpontjában maga a Maple rendszer áll. A szerző kísérletező alkat, lankadatlanul kutatja a rendszer képességeit, feszegeti teljesítményének határait. Mélyre ás az egyes témakörök felkutatásában, nem riad vissza a technikai részletek felderítésétől sem. Hogyan érthetnénk meg a Maple egyszerűsítő eljárásainak működését a kifejezések belső ábrázolásának ismerete nélkül? Hogy szembesülhetnénk a rendszerbe integrált matematikai tudás szintjéről anélkül, hogy betekintenénk a futás során végrehajtott lépések sorozatába? És miközben megismerkedünk a Maple eljárásaival, az eljárások paramétereivel, a problémamegoldás technikáit, a rendszer szemléltetési képességének megismerését mintegy melléktermékként sajátítjuk el.

André Heck könyve alapmű. Olvasása mindenki számára ajánlott, aki meg akar ismerkedni a világ egyik legfejlettebb számítógép-algebrai rendszerével, a Maple-lel.

Maróti György



JGYF KIADÓ
SZEGED

André Heck: Bevezetés a Maple használatába

